PREDICTION OF S-NITROSYLATION POST-TRANSLATION MODIFICATION SITES IN PROTEIN SEQUENCES USING GA BASED FEATURE OPTIMIZATION TECHNIQUE

A thesis

Submitted in partial fulfilment of the requirement for the Degree of

Master of Technology in Computer Science and Engineering

Of

Jadavpur University

By

Aviinandaan Dutta

Registration No.: 140745 of 2017-2018

Examination Roll No.: M4CSE19014

Under the Guidance of

Prof Subhadip Basu

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2019

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that the thesis entitled "Prediction of post-translation modification sites in protein sequences using GA based feature optimization technique" has been carried out by Aviinandaan Dutta (University Registration No.: 140745 of 2017-18, Examination Roll No.:M4CSE19014) under my guidance and supervision and be accepted in partial fulfilment of the requirement for the Degree of Master of Computer Science and Engineering. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other University or Institute.

……………………………………………………

Prof. Subhadip Basu (Thesis Supervisor)
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

Countersigned

…………………………………………………….

Prof. Mahantapas Kundu
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-32.

…………………………………………………….

Prof. Chiranjib Bhattacharjee
Dean, Faculty of Engineering and Technology,
Jadavpur University, Kolkata-32.

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Certificate of Approval*

This is to certify that the thesis entitled "Prediction of post-translation modification sites in protein sequences using GA based feature optimization technique" is a bona-fide record of work carried out by Aviinandaan Dutta in partial fulfilment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering, in Department of Computer Science and Engineering, Jadavpur University during the period of August 2017 to June 2019. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

……………………………………………………………………………..

Signature of Examiner 1

Date:

……………………………………………………………………………..

Signature of Examiner 2

Date:

*Only in case the thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "Prediction of post-translation modification sites in protein sequences using GA based feature optimization technique" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Technology in Computer Science and Engineering.

All information has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Aviinandaan Dutta

Registration No: 140745 of 2017-2018

Exam Roll No.: M4CSE19014

Thesis Title: Prediction of post-translation modification sites in protein sequences using GA based feature optimization technique

…..……………………………………….

Signature with Date

Acknowledgement

I would like to start by thanking the holy trinity for helping me deploy all the right resources and for shaping me into a better human being. I would like to express my deepest gratitude to my advisor**, Prof. Subhadip Basu**, Department of Computer Science and Engineering, Jadavpur University for his admirable guidance, care, patience and for providing me with an excellent atmosphere for doing research. Our numerous scientific discussions and his many constructive comments have greatly improved this work.

I would like to thank **Prof. Mita Nasipuri,** Co-ordinator of CMATER Lab, Department of Computer Science and Engineering, Jadavpur University for providing me all necessary infrastructure and facility to carry on this research.

I would like to thank **Prof. Mahantapas Kundu**, Head of the department, Department of Computer Science and Engineering, Jadavpur University for extending his help to complete this thesis.

I am deeply grateful to my seniors **Mr. Anup Kumar Halder** and **Mr. Soumyendu Sekhar Bandyopadhyay** for their guidance and supervision throughout the project work. Without their enthusiasm, encouragement, support and continuous optimism this thesis would hardly have been continued. I would like to acknowledge their patience in teaching me the workings in the world of Bioinformatics. I would also like thank **Mr. Nirmal Das** for lending the template used in this thesis and for educating me about Mendeley, which has proved to be an indispensable tool in managing all the citations.

Most importantly none of this would have been possible without the love and support of my family. I extend my thanks to my mother **Mrs. Barnali Dutta** and my sisters **Miss Anuleka Dutta** and **Miss Anuska Dutta** without whose forbearance and whole-hearted support this thesis would not have been able to see the light of day.

Lastly, this thesis would not be complete without the mention of a number of wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

……………………………………………..

Aviinandaan Dutta

Registration No: 140745 OF 2017-18

Exam Roll No.: M4CSE19014

Department of Computer Science & Engineering

Jadavpur University

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An omnipresent endeavour in the pursuit of enriching our medicinal knowledge is the identification of specific events that happen as precursors to the onset of diseases. Timely information of such events, also known as biomarkers, are key to early detection of many life-threatening illnesses such as heart disease and cancers. Potentially increasing the success rate of subsequent treatments to a great extent.

1.1 Biological Significance

A living being is a microcosm in itself. It is characterized by a range of biochemical processes that impart to it the unique complexities of life [1]. Series of interactions between biological molecules bring into effect the processes that occur within cells and between cells, which have been established as the functional units of life. Progressive breakthroughs in molecular biology have been increasingly successful in identifying the phases of such interactions thereby facilitating biomarker discovery. Among the classes of biomolecules, proteins act as major functionaries in keeping the biochemical processes active [2], [3]. Acting as catalysts in the form of enzymes and hormones, filling out physically as structural units, enabling specific functions of the body and others.

The three-dimensional structural conformations of a protein macromolecule determine its functions and that its structure is determined from its sequence has been the paradigm [4]. A protein undergoes naturally occurring usually reversible chemical alterations in its structure during or post the various stages of its formation from the genetic material. One such phenomenon is the covalent modifications that occur on the polypeptide chain forming the protein soon after its formation termed post-translational modifications (PTM). Examples of such modifications are addition or removal of molecules from peptides, addition or transfer of functional groups, sugars, lipids or other peptides and cleavage of bonds at amino acid sites among many others. An estimate revealing about 5% of the human proteome to be enzymes involved in catalyzing over 200 types of known PTMs vindicates the hypothesis that PTMs play a pivotal role in making the protein function the way it should to keep the biological systems stable [5]. This is based on the assumption, and a considerable increase in their validations in recent years, that the overall structure and function of a mature protein in a cell is influenced by PTMs. Therefore, the orderly expression of PTMs is critical to the health of an organism. A prime example of a biomarker is the identification and quantification of proteins that are differentially expressed in diseased individuals. It is beyond a reasonable doubt that getting behind the abstraction of the functioning of PTMs is of utmost importance to understand the entire sphere of biological mechanisms a play. Therefore, full characterization of PTMs is an interest with a high priority for the research community. This has resulted in a revised goal in the field of molecular biology to study and document the vast expanse of proteins.

Capturing the entire gamut of proteins into a perspective is a seemingly arduous and challenging task. The vastness of the problem can be gauged from a statistic of the human genome showing it to comprise of a few thousand genes that give origin to more than a million proteins, including multiple variants of the same protein present in the proteome [6], [7]. Experimental techniques that are confirmatory in nature such as Western blotting [8] and Mass spectrometry [9] and others [10] are associated with temporal and cost overheads. Moreover, because of numerous genome sequencing projects, a huge number of protein-coding regions and the related sequences are being identified every passing day [11]. This has resulted in a demand for credible alternative methods for annotation of proteins or those that can at least narrow down the search largely.

With the increase in computational or *in silico* technologies, it is an ongoing attempt to devise algorithms and software that can sieve the vast amount of data automatically. These methods, in general, look for patterns in the data if any that can hint at meaningful information, such as identification of potential post-translational modification sites in this case. Algorithms proposed to date have tried to approach the problem from various angles. They can be broadly categorized into protein primary sequence mining and machine learning models. Multiple facets need to be factored while designing an algorithm for such a problem depending on the approach. While sequence mining methodologies are based on a fairly straightforward intuition they fail to perform well. This evidently points to underlying properties that possibly exist in intricate associations that are not very obvious at the sequence level. Machine learning algorithms as the name suggests aim to learn from the data rather than work on some pre-defined rules. There exists multiple such strategies, which have been used extensively to model complex patterns and they have been shown to perform reasonably well.

1.2 Scope of the proposed work

This thesis was set out to tackle the challenges in designing a binary classification algorithm to segregate sites of a protein sequence into those that have post-translational modification and those that do not. As a technology demonstrator, the focus has been maintained on a particular type of PTM called S-nitrosylation. The reason behind the choice ranges from the consideration of it being a relatively less explored modification among the rest, the scope for improvement in precision to the fact that recently there have been increasing evidence of S-nitrosylation being involved in pathologies. The issues that are well known to affect the performance of such an algorithm at various stages of designing the algorithm has been tackled by making use of existing paradigms of formulating a PTM prediction problem. Greater attention has been paid to the step of optimizing the attributes that can potentially distinguish between a positive and a negative instance of the problem with the help of a meta-heuristic search technique. The resulting performance of the work is compared with other such existing state-of-the-art algorithms and is shown to improve on some parameters. Since the decision making logic of a machine learning algorithm is complex, the exact mechanisms that are at work are quite hard to decipher. The proposed attribute selection procedure aims to alleviate this issue by listing out a subset of the entire set of attributes that results in optimal performance. Thus providing pointers to properties describing the pattern.

1.3 Organization of the thesis

The remaining chapters of the thesis are organized as follows. Chapter 2 provides a background in proteomics analysis. Chapter 3 gives a primer on computational approaches to proteomics. It discusses some optimization strategies that have been proposed in the literature and the challenges associated with them in detail. Chapter 4 is primarily concerned with the design of feature sets and the algorithms developed to generate the classifiers. Chapter 5 discusses the experimental findings and finally paves the road for Chapter 6, which concludes by summarizing the work and sheds light on future scopes.

Chapter 2

Background on Proteomics analysis

A number of biological processes within the cell of a living organism is facilitated by macromolecules called proteins, which form an essential part of its existence. Proteins are responsible for a wide range of functions that is required by the biological system maintain its active state such as catalysis of biochemical reactions, cell signaling, cell adhesion, immune responses, nutrient storage, formation of scaffolds that maintain cell shape and transportation of molecules (including other proteins) between subcellular organelles and across the cell membrane [12].

2.1 Central Dogma of Molecular Biology

All the information required for the cell of an organism to carry out its designated functions is encoded in the genetic material contained in the chromosomal macromolecules or DNA. The protein macromolecule is composed of a chain of amino acids (polypeptide) the length of which can range from tens to thousands of subunits (residues). Construction of a protein is carried out by decoding the instruction from the respective protein encoding region of the DNA known as genes. The Central Dogma of Molecular Biology outlines the flow of genetic information in the cell [13]. Two of its three main parts, transcription and translation (the other one being replication) describe the means by which a DNA sequence specifies the sequence of amino acids in a protein. DNAs are macromolecules that are composed of double-stranded sequence of nucleotides. Consecutive groups of three nucleotides, known as codons, determine the particular amino acids that would occur sequentially in a polypeptide. The relationship between particular codons and particular amino acids has been found to be the same for nearly all living organisms better known as the genetic code. Transcription involves creating an intermediate molecule called messenger ribonucleic acid (mRNA) from DNA. mRNA is similar to DNA in that it consists of a long, specific sequence of nucleotides. The mRNAs are then stripped of unwanted segments before they are translated into proteins by ribosomes that interpret the sequence of codons into their respective amino acids.

2.2 Post-translational modifications

All throughout the various stages of protein formation mentioned above, an aberrant incident in any of the stages results in the introduction of “noise” in the information being propagated. Such modifications are usually ignored and occasionally useful for the correct expression of a protein. Of particular importance, and the area of interest of this work, are the modifications that occur during translation. A protein can be active or otherwise at the time of translation. The alteration between these two states is regulated by certain chemical modifications on the polypeptide chain. This protein modification which is generally referred to as a post-translational modification, can be either co or post its translation from the mRNA and involves the addition of a chemical group, the removal of amino acids from the beginning of the protein or a mutation of an amino acid from one to another [12]. Positions, where PTMs take place in proteins, are called modification sites. These are generally site-specific, which means a particular type of PTM affects only a specific subset of amino acids and are not therefore random in nature. The possibilities of a site being modified is believed to be influenced by a number of factors such as the type of the modification, properties of the amino acids in its neighbourhood, proximity to functional protein sites, the structure of the folded protein molecule, its location in the cell and there are probably others yet to be discovered. PTMs most often than not end up as deciders of cell dynamics such as its function and interaction with other molecules as its occurrence usually results in a change of the protein molecules three-dimensional shape. It has been shown that a protein’s function can be regulated by a PTM by either activating or suppressing it [12].

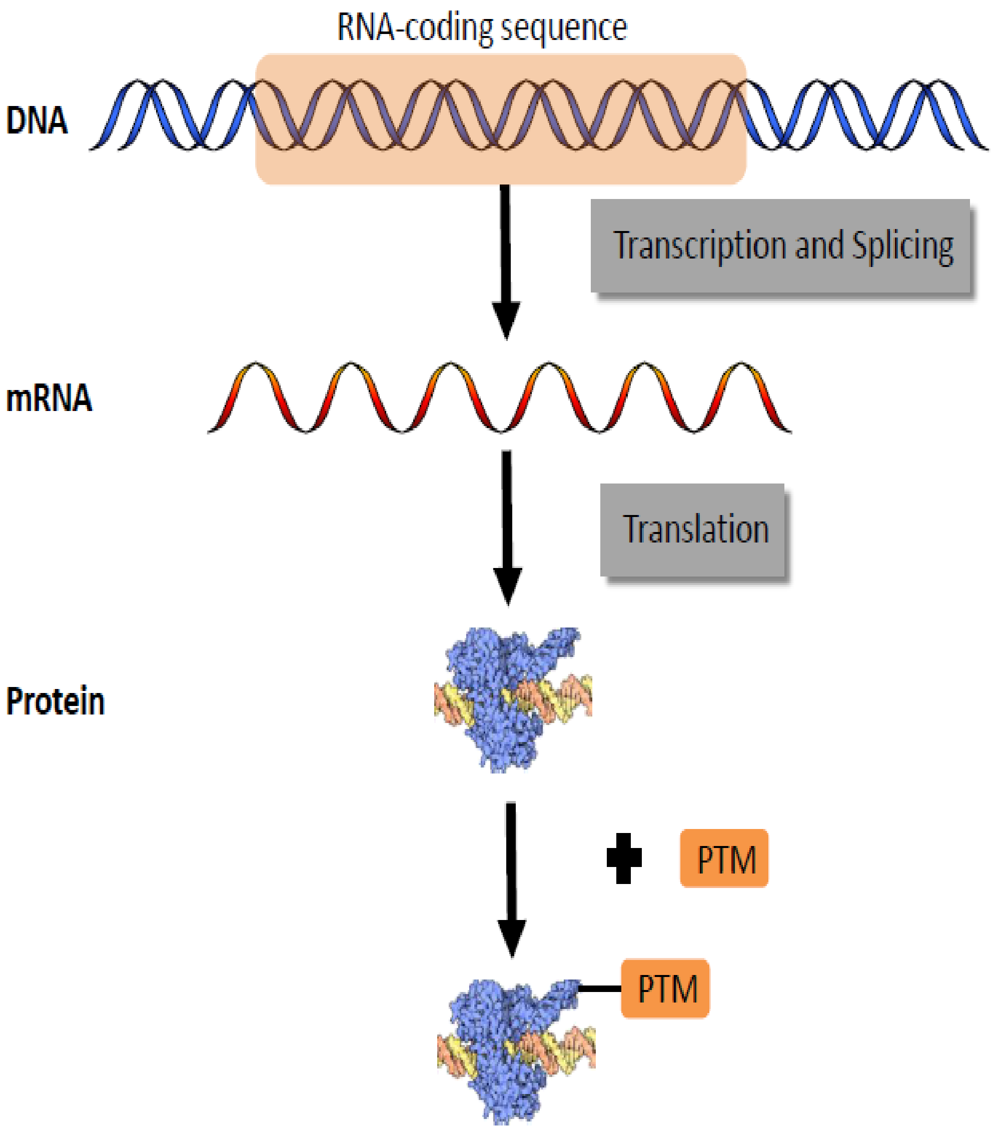

Figure 2.1 - The central dogma of molecular biology [14]

Phosphorylation can be cited as an appropriate example for illustrating the importance of PTMs in the cell and the complex mechanisms that occur during modification. Phosphorylation belongs to the category of the most well-studied PTMs owing to its ubiquitous influence in the workings of a cell, regulating a number of essential enzymes (protein molecules that act as catalysts of biochemical reactions) and receptors (protein molecules that binds to other small molecules such as a hormone to initiate cellular response) [12], [15], [16]. It is the primary mechanism for switching the state of activity of a protein. Thus for many signal transduction pathways in a biological system, phosphorylation of a protein molecule is considered to be necessary. Phosphorylation and its converse mechanism de-phosphorylation are catalyzed by kinases and phosphatases respectively which belong to an enzymatic family of proteins. Phosphorylation includes transfers of a phosphate group from a high-energy donor molecule such as ATP (the biological equivalent of a capacitor) to a substrate of the protein being phosphorylated by a protein kinase. Thereby resulting in a breakdown of ATP into ADP and activation of the substrate by induction of a conformational change in the structure of the protein. Similarly, de-phosphorylation is the removal of the phosphate group by phosphatases and serves as a mechanism to de-activate the phosphorylated protein by removing the phosphate group through hydrolysis. Reversibility of a PTM holds as much importance to the biological system as the forward process.

The PTMs of proteins have been detected by a variety of experimental techniques, which includes the likes of mass spectrometry (MS) [17], liquid chromatography [18], chromatin immunoprecipitation [19], western blotting and eastern blotting [20]. The MS technique is one of the mainstay routes in detecting PTMs in a high throughput manner. At the core of an MS-based experiment is an analytical instrument called a mass spectrometer that measures masses and relative concentration of atoms and molecules. A PTM is characterized in an MS by the variation in the mass of the PTM substrate that is otherwise absent. A new method of detection based on MS coupled with capillary liquid chromatography have resulted in path-breaking advances in enrichment technologies bolstering the quest for confirmatory validation of various PTMs [21]. The last decade has witnessed the identification of thousands of modification sites with great precision and confidence. To date, more than hudreds of PTMs have been experimentally discovered *in vivo*. Through experimentation, it has been observed that PTMs affect protein folding [22], interactions with other proteins [23], protein degradation [24] and hence are key players in cellular regulation including regulation of the cell cycle [25], apoptosis [26], signal transduction [27] from the receptor to the gene among others.

Since proteins play a significant role in cell functions, a disruption in normal protein function can lead to catastrophic effects in the cell and, consequently to the entire biological system. On a similar vein, since PTMs are a major part of the biological mechanism regulating protein function, a disruption in PTM related processes can also have a detrimental effect on the organism. A study revealed about 5% of mutations associated with pathologies to be at known PTM sites, whereas neutral mutations account for only 2% [28].

Aberrations in PTMs hold major weight in the diagnosis of major ailments such as cardiovascular disease [29], multiple sclerosis [30], cancer[31] and many others [32]. Radivojac et al., [33] investigated the role of phosphorylation in cancer and found both gain and loss of a phosphorylation site in a target protein. Death of nigral neurons which is a leading cause of Parkinson's disease has been shown to be aggravated by disturbances to proteolytic pathways set up by ubiquitination (another known PTM) caused by a mutation in the protein *parkin* which is an integral part of its working [34]. Likewise, patients with Alzheimer's disease has been found to exhibit abnormal hyper-phosphorylation of the microtubule-associated protein *tau* [35]. Moreover, mutations of genes at or near the insulin signalling region, which can be PTM impairing, have been shown to be a major factor in Type 2 diabetes mellitus [36].

2.3 S-nitrosylation

One of the most common types of reversible post-translational modifications is the covalent modification of a protein's cysteine thiol (the sulphur atom to be exact) by a nitric oxide (NO) group known as S-nitrosylation [37]. Under physiological conditions, NO is a freely diffusible signalling molecule produced by NO synthases. In addition to its radical nature, the ability of NO to diffuse through cell membranes leads to a wide range of interactions with biological targets in a redox fashion. As a signalling molecule, NO is able to regulate many vascular and neuronal signalling pathways, as well as mitochondrial proliferation [38], [39]. In most cell types, protein targets in close vicinity of the NO synthases may be nitrosylated in place to form S-nitrosothiols by direct communications or through scaffold and adaptor proteins. The enzymatic mechanisms of protein S-nitrosylation are still not clear, however, several enzymes have been demonstrated to facilitate S-nitrosylation or de-nitrosylation reactions. For example, Cu, Zn superoxide dismutase and thioredoxin promote S-nitrosylation, while protein disulfide isomerase is hinted to regulate de-nitrosylation [40], [41].

S-nitrosylation plays a critical role in multiple physiological processes. Many intra-cellular signalling mechanisms are influenced by S-nitrosylation [37], [42]. It also has major say in transcriptional regulation [43], cell signalling [44] and apoptosis [45]. Therefore in a converse logic aberrations of S-nitrosylation in the processes which are influenced by it is associated with the pathophysiology of disorders such as cancer [46], ALS [47], Alzheimer’s disease, Huntington disease, schizophrenia, mental disorders and Parkinson’s disease [48], [49]. Furthermore due to its reversible nature, to make the signalling function of S-nitrosylation even more complex evidence of events where other PTMs such as phosphorylation, ubiquitylation, palmitoylation, acetylation and sumoylation have indulged in crosstalk with S-nitrosylation has been increasingly reported [50]. This comes with an obvious implication of an added dimension to the pathology of diseases.

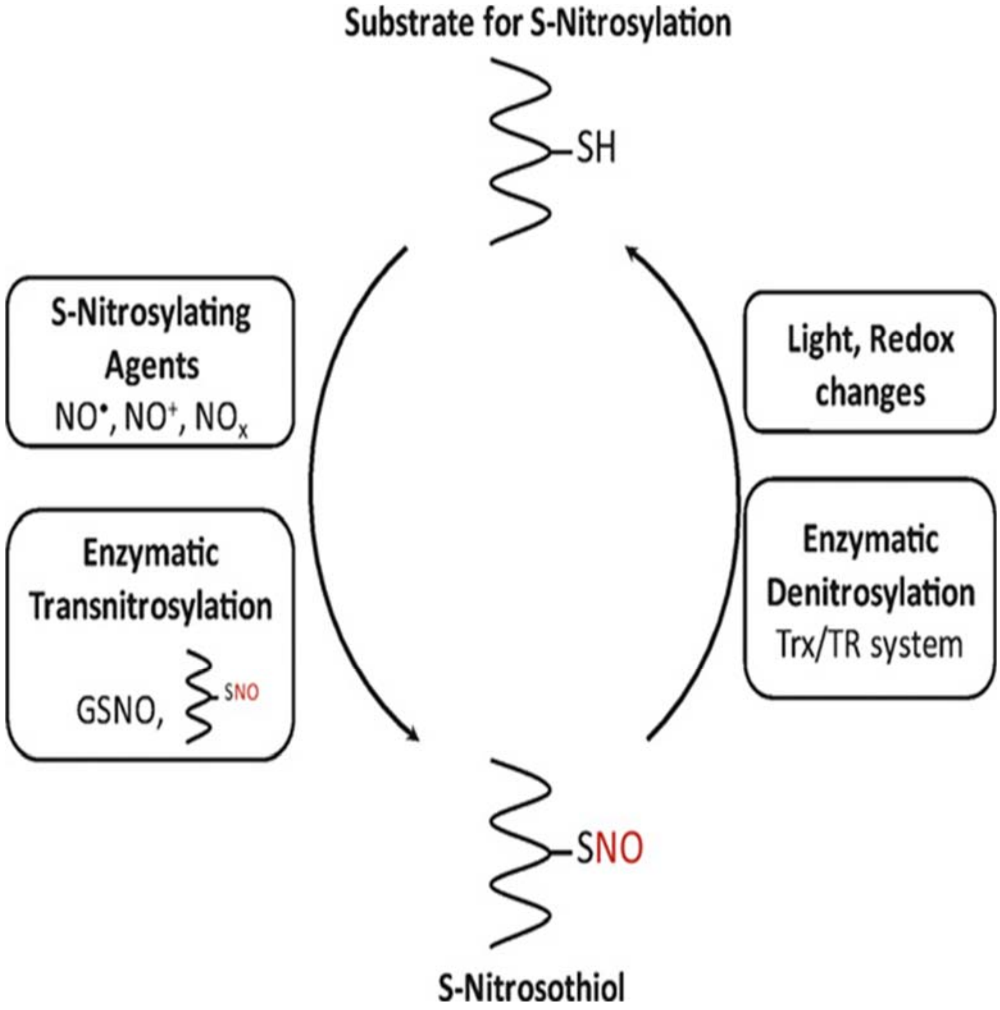

Figure 2.2 - Illustration showing Snitrosylation and de-Snitrosylation of a protein [51]

2.4 Motivation for automation

Keeping the discussion in the previous two sections in mind it can be convincingly agreed upon that need for accurate identification of S-nitrosylation sites in proteins at the earliest shrieks of urgency. Initial attempts to identify candidates on a proteome-wide scale using NO donors as S-nitrosylation agents were mired with false positives due to the lack of in vivo conditions. Conventional experimental identification strategies improved upon the situation by a marking and replacement mechanism widely known as the biotin switch technique [52]. This, complemented with traditional PTM identification techniques such as mass spectrometry has led to improvements in identification of modification sites and has resulted in a proliferation of site-related information curated in various databases [53]. However, even these state-of-the-art experiments are laborious and of low-throughput due to the fact that levels of endogenously nitrated or nitrosylated proteins in the cell is usually low. Efforts to improve efficiency requires great escalation of expense. Moreover, proteomic analyses of nitrosylated sites generated in vivo are usually a challenge due to the low level, dynamic and unstable features of S-nitrosylation. In biological systems, nitrosylated events undergo photolytic degradation [54] and are reduced [55]. In this regard, further efforts are needed to improve the efficiency of current proteomic methodologies. Although much effort has gone into finding patterns such as consensus structural features to describe the specificity of S-nitrosylation based on a large number of datasets from different proteomic studies, accurate prediction of S-nitrosylation sites in proteins still remains a challenge. The prospect of an automatic proteome analysis technique is an exciting proposition given the astronomical rise in computational prowess over the years. Continuous enrichment of results from similar attempts for other types of PTMs provide the required motivation to embark upon such a task.

Chapter 3

Computational approaches

In the early days of molecular biology, the function of a protein was inferred from patterns in external observations of pathological expressions and chemical properties. Much before the sequence of amino acids encoded by the respective gene was revealed. The advent of genomic sequencing and subsequent annotation with high throughput techniques has changed the situation considerably. There has been an explosive increase in the number of known sequences along with residue specific information, which are accumulating in databanks online and are easily accessible. Experimental methods of annotation have not been able to keep up due to their time-consuming nature. Moreover, the fact that the modalities of annotation such as PTM type do not apply to the majority of residues in the primary structure of a protein and at times not even to those that are known to be affected in a different instance cause exploration that would be otherwise unnecessary. This has led to a huge number of functionally uncharacterized genes and proteins thus setting the stage for a computational solution, which comes with an advantage of low cost and fast processing of large volumes of information. This provides supplementary information in narrowing down the search space of potential candidates on a proteome-wide scale and rapidly generate useful information for further experimental investigation. PTM prediction is a valuable tool for guessing the range of possible functions that a protein can be involved in. The knowledge of PTMs that affects a protein is also important for those proteins that have some other features annotated. Further characterization with a predicted PTM increases the chances of discovering a new pathway or a biological mechanism. From the biological point of view, it is important to know which post-translationally modifying enzyme is the source of the PTM for a given substrate protein since this relationship carries a piece of pathway information. Computer-aided prediction of the possibility of a protein's PTM from the amino acid sequence is an important task that is critical to the biological interpretation of proteome data.

3.1 Sequence similarity based approaches

Since the appearance of the first DNA sequences, scientists have sought patterns in them to explain and predict biological phenomena at the molecular level. As a starting point, computational approaches in this problem domain tried to assimilate the unearthed protein sequence data in order to calculate the similarity if any between proteins that go under similar modifications. Tools such as BLAST[56] that computed amino acid sequence similarity were designed to search available databases for target proteins that have high similarity to a specific query protein or peptide to gather ideas about its attributes. Observations found conserved regions in sequences which led to the definition of consensus patterns. Subsequently, the attribute annotations of the target proteins thus found were used to infer the attribute for the query protein. These patterns obtained from such straightforward sequence mining models, although confirming the direction of research, were not very specific and were not a strong enough decider. Also, they lacked the capability of generalization to unknown proteins that did not exhibit sequence similarity with any of the known proteins. But the mere presence of them indicated deeper underlying associations between other features of the proteins. This has been subsequently verified in experiments [57]–[60].

3.2 Machine learning models

The next step up in complexity included the identification of features possibly having correlations that would together form patterns discriminating the sequences of interest from the rest. Query sequences could then be annotated using decisions based on similarity scores of the sequence features to the pattern identified. One of the first examples of such features were weight matrices[61], which allotted scores based on the probability of an amino acid residue occurring at a position in a subsequence. This allowed a much more diverse description of patterns. Going even further in complexity, as more features started getting included in the list of probable pattern descriptors it led to a proportionate increase in the complexity of the computational models required to process these. With the introduction of machine learning concepts, computational pattern recognition received a major boost that broadened its scope hugely [62]. It was now possible to extract feature associations that were not obviously comprehensible on manual exploration automatically.

Machine learning is a computational discipline in the field of artificial intelligence that involves generating inferences about a collection of data from its underlying statistical properties, by building a mathematical construct using them, and subsequent use of the generated inferences for decision making without depending on any explicit instruction [62]. A model is an abstract representation of the problem of interest that describes the intricate relationships between observations and the apparently hidden causes that give rise to such observations. Given a real world model, a machine learning algorithm attempts to “learn” or discover the pattern that describes the model by replicating it into a mathematical model based on a number of instances of observations that is given as input to the algorithm. Therefore, a general machine learning based computational pattern recognition approach to solving a problem begins by designing a problem model and an appropriate learning algorithm. This is followed by the algorithm learning the model using an input data (called training data) that instantiates the model. Generalizing from experience is a core objective of a learner. If a learning algorithm is able to perform accurately on new, unseen data after gathering experience from a training data it is said to generalize well. To look at it from a theoretical point of view, the learner that is able to generalize well is able to build a general model about the space of distribution the training samples belong to. The learned model can then be used to make predictions about unknown (test) data by finding the probability of match to the discovered patterns. Various strategies of creating such a mathematical learning model have been proposed in the literature over the years. These differ in their approach, the type of data they can handle and the type of task or problem that they are intended to solve and can be broadly grouped into three categories:

a) Supervised and semi-supervised learning
b) Unsupervised learning
c) Reinforcement learning

Among these, supervised learning most suits the problem definition being worked upon. Supervised learning is the family of machine learning algorithms that takes supervised data instances (data which have been labelled to belong to a particular class) as input, tries to find the patterns ingrained in the data that best describe a class, capture the pattern in a model and finally use the model to classify (predict the class) of a new data instance. In the mathematical model, each training example also called a pattern instance, is represented by an array or vector of values for the features that express the pattern instance. The outputs or class labels are provided as another vector. The training data in its entirety is represented as a matrix (vector of vectors). Through iterative optimization of an objective or loss function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. The optimality of this function learned by the algorithm determines its performance on data that it has not experienced during training. Mathematically this concept can be explained as, given an unknown function $\boldsymbol{f}: \boldsymbol{A} \rightarrow \boldsymbol{B}$ mapping input examples $\boldsymbol{a} \in \boldsymbol{A}$ to output labels $\boldsymbol{b} \in \boldsymbol{B}$, along with training data $\boldsymbol{X} = \{(\boldsymbol{a_1}, \boldsymbol{b_1}), \ldots, (\boldsymbol{a_n}, \boldsymbol{b_n})\}$ that is assumed to represent accurate examples of the mapping, produce a function $\boldsymbol{h}: \boldsymbol{A} \rightarrow \boldsymbol{B}$ that approximates as closely as possible the correct mapping $\boldsymbol{g}$. The loss function which is used for calculating the approximation function by the algorithm assigns a value to the error resulting from producing an incorrect prediction.

In the context of the present problem, the input model can be described as a two-class classification task where a suitable representation of a protein substrate site is said to belong to one class (let's call it positive) if it is found to be modified by a PTM from existing PTM data of known proteome, else to the other class (let's call it negative). Several PTM specific predictors have been published in the literature for a number of different types of PTMs [63]. In addition to these, a number of meta-predictors have also been designed that keep multiple PTMs in scope to try and identify PTMs for whole proteomes. The AutomotifServer [64] and PTMProber [65] is an example of such a meta-predictor that implements different models for each of the PTM types and uses a consensus scheme to arrive at the final decision. These methods differ in multiple aspects such as the training and test datasets used, the ratio of positive vs negative samples, feature selection strategy, feature vector representation, machine learning algorithm used and the specificity obtained from the design on the test set and more. The ideal predictor is characterized by its ability to confidently predict potential PTM sites which are also accurate. In the practical scenario though, more often than not a compromise has to be struck upon the number of positive class predictions obtained that are actually positive and the number of actual positive examples that are recognized as positive by the predictor when tested on a portion of the known data kept hidden from the predictor during training. Other issues affecting the algorithms range from the complexity of the mathematical models to the constant updation of available databases thus creating room for a lot of improvements. The above discussion points to several concerns that need to be taken care of while designing a PTM predictor:

1) Quality, reach and sources of the dataset being used.
2) Engineering of features and selection of the ones that are finally used to build the predictor model.
3) The prediction algorithm to be used and any assumptions it makes about the problem.
4) Evaluation schemes for analysis of the prediction results.

The following sections are aimed at providing a brief insight into the details of designing a PTM predictor and the challenges associated with it.

3.2.1 Data sources

Quality of prediction is largely influenced by the diversity of the data that it models. Higher the diversity, higher the chances of capturing all underlying patterns. It is difficult to estimate the amount of data that would be required by the learning algorithm to build a reliable predictor with sufficient confidence since exact features that can accurately explain a pattern is not known. An increase in the number of features being considered increases the complexity of possible correlations thereby subsequently increasing the requirement for training examples in order to effectively capture those correlations. In addition to data from *in vitro* experiments on incubations of a randomized peptide library with a given enzyme, databases recording various properties of proteins are available at present. A typical record in the database contains a subset of protein identifier, sequence position number of the modified residue, type of the residue, type of modification, attached chemical group, modification enzyme, citation information and characteristics of the experiment as fields. There are PTM specific databases that aggregate low as well as high throughput experimental observations for a particular type of PTM from the scientific literature. PhosphoSite [66] for phosphorylation, OGlycobase [67] for glycosylation are to name a few of the popular ones. The key repository though is the internationally managed UniProt knowledgebase [11] that is a comprehensive database of a number of protein properties and functions along with PTM annotations. The data in the UniProt are collected from information resources and bibliographic sources. The dbPTM [68] database is an information repository exclusively for PTMs retrieved from several databases including UniProt as well as data extracted from publications. The annotated sequences retrieved from these databases are used to assemble the dataset that is mined by the predictor.

3.2.2 Data preparation

Proper representation of the data that is to be given as input to the learning algorithm for its training is mandatory for it to build a model that is equally representative of the various underpinnings in the data. The major considerations in this direction are discussed in the following sections.

3.2.2.1 Labelling

For a PTM in consideration, the sequence positions of target residues that are annotated to be modified by it are denoted as positive instances. The following step of labelling the negative instances is not so obvious as it is hard to gain exhaustive surety of a residue not being modified under any biological condition or environment. Therefore it is difficult to get around the ambiguity regarding the labels. The reason for this is, the reactions resulting in PTMs are stochastic in nature. Modifications are determined in varying conditions in different experiments. Therefore, the available data does not contain exhaustive information on the occurrence of a PTM on a particular protein. This leaves place for alternate interpretations of the negative patterns. At present, the construction of negative sets has no uniform standard. There are quite a few strategies that have been used by researchers to circumvent the situation under certain assumptions. The most common of them is to sample positive and negative instances in a 1:1 ratio or in a 1:x ratio where ‘x’ is a reasonable multiple [69]. A strategy that has been used frequently marks all remaining target residues of a PTM in a particular protein that has not been found to be modified through experimentation as negative instances [70]–[73]. The rationale behind this assumption is that, it is better to use the residues from the proteins experimentally inspected that have not shown any modification albeit in a specific set-up, than residues from proteins that have not been studied. Plewczynski et al. [64] extracted instances that have not been experimentally found to be modified by any PTM at random and labelled those as negative instances. Blom et al. [74] generated an augmented set of samples by using predictions from a learning algorithm designed using the majority approach and selected the ones that were most isolated. Another strategy that has found wide acceptance is the bootstrapping method of generating $\boldsymbol{N}/\boldsymbol{P}$ bins of randomly selected negative instances where $\boldsymbol{N}$ is the total number of negative instances and $\boldsymbol{P}$ is the total number of positive instances, training a similar number of learning algorithms and taking the majority prediction as the final decision [75].

3.2.2.2 Training sample representation

Once the data to be looked into has been decided upon, it must then be formulated into a numeric representation that can be exploited by the learning algorithm. The most obvious way forward is to consider the protein's amino acid sequence for its representation. However, it has been found that a particular type of PTM occurs at a specific amino acid residue or a subset of residues [76]–[78]. It has also been observed that the mere presence of a target residue for a particular PTM in a protein's primary structure does not always imply its modification [79]. Prompted research in this direction due to this observation has led to the discovery of conserved regions in the sequences of modified proteins also called consensus sequences or motifs. This points to the presence of some common attributes in these conserved zones that are essential for its function and thus has resisted evolutionary changes [79]. It is these regions therefore, which are of most importance and worth investigating. Further studies have strengthened this belief. Crystallization studies indicate that during phosphorylation, a region between seven and twelve residues in size surrounding the acceptor residue comes in contact with the active site of the enzyme that catalyzes the modification [77]. Based on large sets of experimentally verified phosphorylation sites, Blom et al. [74] found certain residues to be expressed more than others in the context of phosphorylation sites. Their exploit was subsequently validated when some of the sites they predicted to be potential candidates for phosphorylation exhibited homology. Existence of possible motifs for phosphorylation has been confirmed in a number of other experiments [57], [80], [81]. Methylation of the arginine residue in a number of proteins was shown to be on an arginine-glycine-glycine (RCG) motif [82]. S-nitrosylation of cysteine residues were hypothesised to be flanked by acidic and basic amino acids[76], have aromatic side chains [83] or be embedded into a hydrophobic area [84] which suppressed or favoured the modifications. These pieces of evidence led to the conclusion that the event of post-translational modification of a particular target residue is influenced by its immediate neighbourhood in the protein sequence. Following these revelations, experimental strategies and computational approaches started incorporating local sequence information into their methods for prediction [72], [85].

To set the reference frame for considering the neighbourhood, an alphabet of 20 symbols is taken for the 20 individual amino acid residues each. A training sample instance $\boldsymbol{I_k}\ (\boldsymbol{k = 1, 2, \dots, P + N})$ is represented by a subsequence window of $\boldsymbol{2m + 1}$ residues from the protein sequence at a time with the site being considered for modification at the centre which can be expressed as

$$\boldsymbol{I_k = R_{-m}R_{-(m-1)}R_{-(m-2)} \dots R_{-1}R_0R_1 \dots R_{+(m-2)}R_{+(m-1)}R_{+(m)}} \qquad (1)$$

where $\boldsymbol{R_0}$ is the target residue at the centre, $\boldsymbol{R_{+m}}$ the $\boldsymbol{m^{th}}$ upstream and $\boldsymbol{R_{-m}}$ the $\boldsymbol{m^{th}}$ downstream amino acid residue from it. A binary label indicating the occurrence of a PTM type is associated with each instance. Such a $\boldsymbol{2m + 1}$-tuple representation of a training sample can then be labelled as:

$$\boldsymbol{I_k} = \begin{cases} \boldsymbol{1}, & \boldsymbol{if\ R_0\ is\ positive} \\ \boldsymbol{0}, & \boldsymbol{if\ R_0\ is\ negative} \end{cases} \qquad (2)$$

All positively and negatively labelled instances together form the training dataset. The value for $\boldsymbol{m}$ varies from one modification to another due to the different physical natures of each and is generally decided by inputs from domain experts or through experimentation. For example, a very short window would not sufficiently describe a chemical or structural space while increasing the window length comes with an overhead of added complexity and possibly hinder performance rather than enhancing it.

Having decided on the reference frame for a site of interest, to establish a useful predictor model the training samples now need an effective mathematical expression of its features that can reflect their intrinsic correlations and form determining patterns with respect to a class. A number of concerns arise and need to be factored while designing a representation to render it an effective collection of markers that is able to exhibit the intricacies in correlations among the instances of a class that reduce ambiguity as much as possible and determine a good separation.

PID	Cysteine Position	Subsequence	Window Size
P60202	227	NLLSI**C**KTAEF	11
P60202	227	SNLLSI**C**KTAEFQ	13
P60202	227	GSNLLSI**C**KTAEFQM	15
P60202	227	GKVCGSNLLSI**C**KTAEFQMTFHL	23
P60202	227	PGKVCGSNLLSI**C**KTAEFQMTFHLF	25
P60202	227	FPGKVCGSNLLSI**C**KTAEFQMTFHLFI	27
P60202	227	AFPGKVCGSNLLSI**C**KTAEFQMTFHLFIA	29

Figure 3.1 – Window representation

3.2.2.3 Feature extraction

Sequence motifs which are the substrate of activity in proteins can be complex in the sense that correlations between residues along with their specific positions in the sequence may play an important role. The amino acids in the neighbourhood of a target residue may not contribute independently to determining the modification of a site. This means that a simple representation of the sequence windows with their raw character values or linear weight matrices based on motifs may not hold enough information to help the model separate positive sites from the negative ones. Two positions can only make independent contributions, and complex rules that possibly factor positional correlations which are required for obtaining acceptable levels of performance cannot be taken into account [74]. This raises the possibility of the presence of other potential consensus motifs based on features other than the amino acid sequence position.

This presents the opportunity to investigate the vast composition of physical and chemical properties of each of the 20 amino acids as well as other biological properties of the protein in consideration that decides specificity and diversity of its structure and function. Protein sequence information is the most logically relevant feature since it represents the evolutionary forces responsible for conserving functionally useful motifs and have been generally found to be the most distinguishing one too [77], [78]. Thus protein sequence information is considered in almost all predictors. Orthogonal binary coding, amino acid frequencies or amino acid composition [86] are used to represent protein sequence windows as numeric vectors. Since such a discrete representation may cause loss of the positional information, positional weighted matrix [61], position weight amino acid compositions [87], position specific amino acid preference [64], amino acid pair composition [88] were proposed to extract the position information of amino acid residues in the neighbourhood of a target residue. Chou et al. [88] introduced pseudo amino acid composition which proposed a vector representation of proteins that were without significant sequential homology to other proteins and have found wide application in molecular biology and have been subsequently used in a number of biological predictors. Along with sequence information, the identity of a protein is also shaped by the composition of its constituent amino acids. Physicochemical encoding have been shown to be particularly suited for sequence windows [89]. The AAindex is a database of numerical indices representing various physicochemical and biochemical properties of amino acids and pairs of amino acids. Currently, 566 amino acid indices are released in the AAindex1 database. These indices cover extensively various formulations of amino acid properties that have been proposed in the literature over the years and have been used as features in the designing of a number of biological predictors [90]. The structure of a protein is the primary decider of its accessibility. A side-chain of amino acid that undergoes PTM prefers to be accessible on the surface of a protein. Therefore, structural information is potentially a significant feature. However, structural information is hard to ascertain due to temporal and cost constraints associated with experiments. Some of the structural features that have been proposed are accessible surface area [91] which quantifies the percentage of the solvent-accessible area of each amino acid on a protein and B-factor, a measure a protein’s crystal structure that reflects the fluctuation of atoms about their average positions and provides important information about protein dynamics.

3.2.2.4 Feature selection

The capability of a machine learning algorithm is greatly influenced by the choice of features used to represent the data that it is trying to model. Presence of redundant or strictly correlated features imparts unnecessary complexity to the data which correspondingly raises the complexity of the model and possibly dampens its generalization power. It also increases the required number of training examples by the model to capture the complexity [62]. This is an exact appeal to the present situation.

Since the features belong from different sources they are possibly noisy and redundant. Building a predictive model using all the features is fraught with a high computational cost. Therefore, it is imperative to select the ones that would benefit the learning algorithm most. From a biological perspective, a systematic comparative analysis of the best performing features so found would help in the understanding of underlying mechanisms of the biological event being studied. For example, on analysis of combinations of four features (amino acid composition, accessible surface area, position weighted matrix and physicochemical properties) for designing an S-nitrosylation predictor, Lee et al. [72] found amino acid composition to be the most influential individually, and more effective when combined with position weighted matrix. While the performance was reduced when all the four features were considered together. Such an exhaustive analysis would not have been possible if the features being considered were not four but in hundreds. Also, the influence of a particular feature depends on the PTM type. This is due to the reason that different PTMs occur in different biological contexts. Therefore, a feature selection strategy is essential for finding the optimal or near-optimal features of a particular type of PTM.

A feature selection strategy is a combination of a search technique over the space of all possible feature subsets with an evaluation measure which scores the different feature subsets. A naïve way to do this is an exhaustive evaluation of all combinations and selection of the one with the highest predictive performance. This is computationally infeasible for large subsets due to its exponential complexity ($n\, features\; =\; 2^n\, combinations$). A number of works have subsequently explored the use of heuristics to define a semi-optimal selection. Two of the most prominent examples of such an approach are the forward selection and backward elimination techniques [92]. Forward selection starts with an empty feature set and keeps adding features one at a time that results in most improvement in the evaluation measure while backward selection starts with the set of all features and keeps removing features individually that most improves performance. These methods though optimal at each stage are unable to analyze all interactions between individual features thereby restricting the search. Both forward selection and backward elimination techniques of feature selection have been used extensively in designing biological models. A different direction of approach is the use of randomness. Randomized algorithms use randomized or probabilistic steps for selecting the subsets. The Relief algorithm is a notable one in this sphere. It assigns weights to features based on the performance of randomly sampled instances. Improving on the uncertainty of complete randomness genetic algorithms (GA) [93] use a guided randomness approach modelled on the concepts of natural evolution and survival of the fittest. It iteratively evolves a population of possible solutions devising strategies to explore and exploit and the search space. By introducing an element of guidance with help of operators that mimic the survival of the fittest concept, it aims to find the optimal or semi-optimal solution. GAs have been shown to be an extremely potent search and optimization technique and several applications have deployed various avatars of it over the years [94]–[97] for the simplicity of the working principle.

All the feature selection techniques discussed above and a large number of others can be grouped into three broad categories based on the way selection is carried out (independent of the learning algorithm or not) namely filter methods, wrapper methods and embedded methods that combine both filter and wrapper methods. Filter approach selects feature subsets based on the intrinsic statistical properties of the data defined by the feature in a predictor independent way. In contrast, wrapper methods first select a feature subset by some technique and use the performance of the learning algorithm as the evaluation measure for the selection [98]. This implies that the actual model performance plays a crucial role in selecting optimal feature subsets. This approach has been shown to produce better subsets than the filter methods with a tradeoff for computational time and complexity. Meta-heuristic algorithms such as GAs, Particle Swarm Optimization, Ant Colony Optimization and Simulated Annealing among others are applied as wrapper methods to implement the search. An embedded approach attempts to combine the best characteristics of filter and wrapper methods. It generally involves a preprocessing step by a filter method for a wrapper method. In this thesis, an instance of the embedded approach has been used as the feature selection technique. Clustering of the features followed by a two-stage genetic algorithm has been proposed.

3.2.3 Predictor algorithm

The choice of a predictor depends on the type of data it can handle (continuous, discrete, statistical distributions, homology), classification transparency it can provide and the number of required resources for an acceptable level of performance by the model which is also robust and scalable. None of the existing methods is superior to all others for all types of data, but each of these methods has its strengths and weaknesses determining the scope of its applicability. Classification transparency of a model is an important concern from the perspective of biologists since a black box model is incapable of revealing the steps that lead to the actual prediction decision, thus of little value in understanding the underlying biological process. Therefore a model that can chalk out the path taken in the decision-making process such as decision trees and random forests are preferred over a model the working of which is difficult to decipher such as Artificial Neural Networks and Support Vector Machines. However, transparency and performance share a conflicting relationship yet and there is generally a tradeoff.

Sequence homology-based predictors such as the PTMProber [65] deploy sophisticated sequence matching techniques between the query sequence and the annotated protein sequences in the database. Statistical supervised machine-learning models such as the Bayesian methods and Decision Trees have been known to be efficient classifiers. Both these methods are transparent predictors, the rule-based disjunction of conjunction representation of a decision tree and the probabilistic inferences of the Bayesian methods can be followed by hand. While the assumption of non-existence of any correlation between features is a deterrence for Bayesian methods, decision trees have been used extensively. Charpilloz et al. [99] designed a motif based decision tree as a predictor which uses a category of motifs with different similarity properties as nodes for the tree and genetic algorithm to search for suitable candidates in the protein sequence that fit the motifs. The pSuc-Lys [85] uses an ensemble of random forests to predict succinylation in lysines. The computational cost of training these type of models is relatively low but their performance tends to decrease with increasing complexity of feature correlations. The machine learning methods SVM and ANN can model complex correlations between features better than other methods and have been deployed as predictors by many authors. The AMS [64] uses a simple SVM to predict sites for multiple PTMs while Wang, Liu &Wang [73] uses a SVM with two self-designed kernels to realize a multiple predictor. The updated versions AMS3.0 [69] uses multiple ANNs optimized for different evaluation measures using variations of consensus between multiple ANNs. A number of biological predictors use machine learning due to their capability of modelling complex correlations.

3.2.4 Evaluation scheme and metrics

Evaluation of the performance of a predictor with standard metrics is necessary to gauge the extent of its ability to model the problem. Standardization is required to compare it with other prediction tools. In order to evaluate its performance, a predictor needs to be tested on a set of data that it has not learned on. This is achieved by holding out a portion of the data to be used for testing purposes.

Among the several existing measures that reflect different aspects of classification performance, four of them are most relevant in the context of prediction of PTM sites namely accuracy, sensitivity, specificity and Mathews correlation coefficient (MCC) [100]. For an instance in the testing set, the model gives the probabilities of it belonging to each class as output. A user-defined threshold then decides which class should be labelled to it. It is possible to compare the class label predicted by the model with the actual class label of that instance. The number of cases possible over the entire set can be surmised into four scenarios in form of a confusion matrix.

		Actual label	
		Positive	Negative
Predicted Label	Positive	True Positive(TP)	False Positive(FP)
	Negative	False Negative(FN)	True Negative(TN)

The accuracy of the classifier measures the proportions of correct predictions made and can be expressed as:

$$\frac{Number\ of\ true\ positive\ instances + Number\ of\ true\ negative\ instances}{Total\ number\ of\ instances} \quad (3)$$

Sensitivity is a measure of the proportion of positive sites that were correctly identified. Given as:

$$\frac{Number\ of\ true\ positive\ instances}{Total\ number\ of\ positive\ instances} \quad (4)$$

While the measure of the proportion of negative sites that were correctly identified is referred to as Specificity:

$$\frac{Number\ of\ true\ negative\ instances}{Total\ number\ of\ negative\ instances} \quad (5)$$

A good predictor should be able to provide reliable answers for both positive and negative cases. The ideal predictor is characterized by a high true positive prediction rate (sensitivity) and a very low false positive prediction rate (1-specificity). Any predictor is always a compromise between these two oppositional requests. A low threshold results in high sensitivity but is rife with false positives. On the other hand, a high threshold warrants a low false positive prediction rate for a trade with the true positive prediction rate. In a practical context of an experimental laboratory and for proteome scans, the achievement of low false-positive prediction rates becomes more important for the predictor for the recognition of possible cases in feasible time. The area under the receiver operating characteristic curve (AUC), gives a statistic that tries to capture these two measures in a single number. IT calculates the area under the curve formed by plotting values of sensitivity and specificity at different decision thresholds. Taking care of the inherent class imbalance in PTM data becomes all the more important here as a disproportionately large negative set is bound to flood the user with false positive instance predictions by a low-specificity predictor. The Matthews' coefficient of correlation is widely used as an unbiased performance estimator:

$$\frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \qquad (6)$$

MCC gives a balanced measure of the positive and negative instances. For example, a model predicting all instances to be positive correctly predicts all instances which are actually positive but this is of no use practically. The MCC would reflect this since it takes into account all scenarios of prediction outputs. A perfect predictive performance has an MCC value of +1 while a random prediction has a value of 0. In the case of a total misprediction, it has a value of -1. Also of importance is the F1 score that gives the harmonic mean of the metrics of precision and sensitivity where precision is

$$\frac{Number\ of\ true\ positive\ instances}{Total\ number\ of\ instances\ predicted\ as\ positive} \qquad (7)$$

And therefore F1 stands as,

$$\frac{2TP}{2TP + FP + FN} \tag{8}$$

A learning algorithm may accept a set of parameters that need to be tuned for optimal performance. An evaluation scheme is used to tune the different parameters of the model being built. Using the test set to tune the parameters would amount to a biased evaluation. Therefore, there is a need for an additional split of the data for tuning or validation. Various splitting strategies exist among which the popular ones include the k-fold technique, stratified folds, and leave one out validation also known as the jackknife test. The most popular due to its lower computational demand, k-fold cross-validation is used to verify the achieved strength of the training by weeding out any bias that may have been caused by random sampling of instances into training and test set. The entire dataset is divided into k splits based on a chosen strategy. Now, k times $k - 1$ splits are used as training data and the remaining split as test data. The metric scores averaged over k iterations are then used as the final result.

3.3 Genetic algorithm

Section 3.2.2.4 introduced genetic algorithms (GA) as a guided random search and optimization technique for obtaining good semi-optimal solutions to the feature selection problem. Feature selection can be visualized as an optimization problem, that is a search over all possible combinations of features for the subset that provides the best performance. GAs were proposed as a computational procedure that tried to mimic the phenomenon of adaptability so prevalent in nature [101]. Based on the Darwinian theory of evolution of organisms by the “survival of the fittest”, a GA operates on a population of possible solutions from the search space, each of which is encoded in a specific manner and termed as a “chromosome”. Each constituent solution variable (for example a feature) of an encoding chromosome is called a “gene”. The quality or “fitness” of each chromosome is measured using some objective function (the performance of a machine learning model can be thought of as an example). With the help of special genetic operators, “crossover” and “mutation” the GA aims to create a population of new, better chromosomes from the randomly generated initial ones. This is carried on over successive generations to find the optimal solution until a stopping condition is reached. Increase in fitness of the chromosomes over the generations is ensured with the help of a stochastic sampling procedure called “selection” which results in a more directed strategy than a purely random search. Compared to other gradient-based deterministic search techniques [102] a stochastic strategy aided by its randomness is able to avoid convergence to a local optimum solution which plagues the gradient-based techniques. Exploration and exploitation are two competing events that pose a challenge to the genetic algorithm in its search which needs to be considered during its design in order to reach as close as possible to the global optimum solution in feasible time. Exploitation ensures the width of the search or the number of possibly good solutions in different regions of the search space that can lead on to the best solution. While exploration, as the name suggests, refers to the investigation of the neighbourhood of a solution with an expectation that possibly better solutions exist within its vicinity [103]. The balance between exploration and exploitation is controlled by a number of parameters. Other than those that are specific to implementations of genetic algorithms for a particular problem, the population size, crossover rate and mutation rate is common to all and needs to be mentioned. Exploitation of fitter individuals is determined by the selection operator. The crossover and mutation rate determines the rate of exploration. The population size generally determines the reach of the algorithm in a single generation and is generally constrained by the availability of resources. The next few sections provide a discussion on the major modules that constitute a genetic algorithm in the context of a feature selection problem.

3.3.1 Encoding

In order to apply the genetic algorithm, each candidate solution in the search space must be parameterized and encoded as a string called the chromosome. In the feature subset selection problem a solution can be coded as a $\boldsymbol{n}$-bit binary vector where $\boldsymbol{n}$ is the number of features. A chromosome then represents a subset of features, the presence(1) or absence(0) of the j-th feature determined by the value of the j-th bit ($j = 1,2,3 \dots n$).

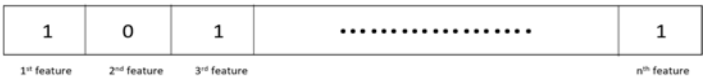

Figure 3.2 – A chromosome for a feature selection problem

3.3.2 Initialization of population

The algorithm manipulates a set of solutions at a time, also called the population. The size of the search space guides the decision on the number of chromosomes $\boldsymbol{m}$ to form a population. The chromosomes of the initial population are generated by setting the bits of the vector to 0 or 1 randomly or according to some heuristic. Often, the initial population is generated in as diverse a way as possible. If the size of the population is too small it may fail to represent enough diversity and the algorithm may converge to a solution prematurely while a very large population size would take a long time for convergence.

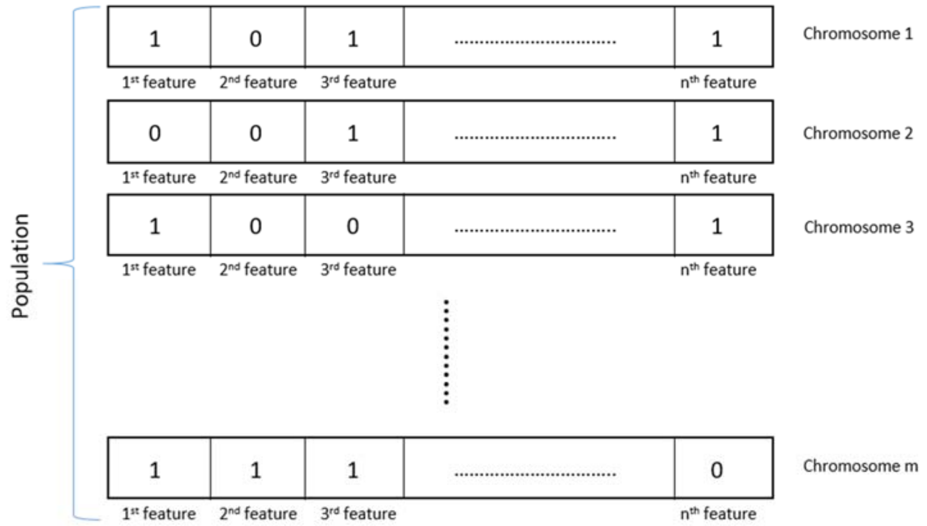

Figure 3.3 – Illustration of a population in a genetic algorithm

3.3.3 Fitness evaluation

For each chromosome $\boldsymbol{c}$ in the population, the learning algorithm is trained and tested using the data described by the subset of features represented by the chromosome. One of the evaluation metrics used to express the performance is then used as the fitness score $\boldsymbol{f}[\boldsymbol{c}]$ of the respective chromosome.

3.3.4 Selection

The selection operator decides which chromosomes from the population would participate in creating the next generation based on a strategy. Usually, the fitness of a chromosome plays a major role in deciding its chances of getting selected as the "parent". This selection operator is responsible for incorporating the concept of Darwin's "survival of the fittest" into the genetic algorithm. Essentially, the better fit chromosomes contain good features and their exploitation along with other well fit chromosomes should possibly lead to better solutions and convergence to the optimal solution in succeeding generations. The selection operator may be parameterized to establish a control on the selection pressure which is necessary for maintaining the balance between exploration and exploitation. A very strong selection pressure would cause the algorithm to converge early without much exploration while a very low selection pressure would cause the algorithm to wander around randomly without any direction. The Roulette Wheel Selection (RWS) method is a very popular choice for the selection operator. The RWS attaches to every chromosome $\boldsymbol{c}$ in the population a probability value $\boldsymbol{p}[\boldsymbol{c}]$ of its selection which is proportional to its fitness $\boldsymbol{f}[\boldsymbol{c}]$.

$$p[c] = \frac{f[c]}{\sum_{c=1}^{m} f[c]} \tag{9}$$

Where $\boldsymbol{m}$ is the population size.

The RWS algorithm can be described as,

```
Input:
Population
Procedure:
totalfitness=0
for c in population
   {
      totalfitness+=f[c]
   }
generate a random number x€[0,totalfitness]
sum=0
for c in population
   {
      sum+=f[c]
      if sum>x
      return c
   }
Output:
Selected chromosome
```

Several selection methods have been designed till date [104] among which Roulette Wheel Selection, Stochastic Universal sampling and the tournament selection are very widely used.

3.2.5 Crossover

Once the parent chromosomes are selected from the population, the crossover operator is used to exchange the information contained by them for further exploration. The idea is to combine the good parts of one parent with the good parts of another to form an “offspring” chromosome that is better than each of the individual parents. Traditionally crossover methods decide on a single or multiple crossover points along the length of the parents by some strategy, dividing them into segments, which are then exchanged. The crossover strategy used plays an important role in balancing exploitation with exploration or vice-versa by causing disruptions in the search direction of the genetic algorithm [105]. A number of crossover strategies have been devised such as the single-point crossover, 2-point crossover, k-point crossover, ordered crossover, etc. which vary in their capability to maintain an order and the amount of disruption they cause. The uniform crossover method has been shown to perform well in large scale feature optimization problems [105]. It considers each gene (bit position) as a segment and exchanges them randomly.

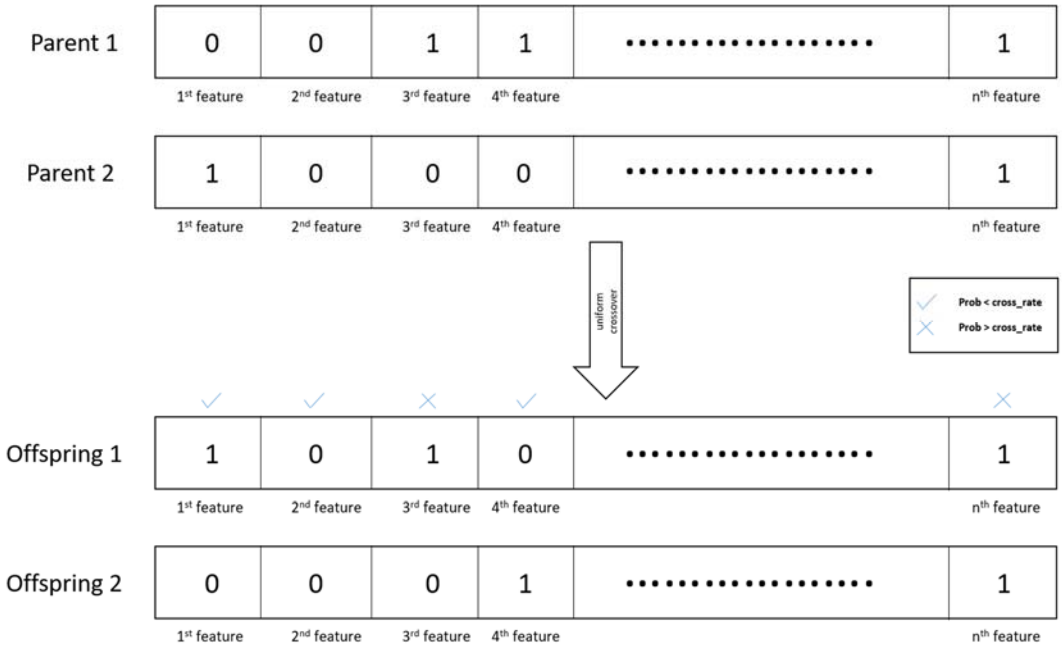
Parent 1
0 0 1 1 •••••••••••••••••• 1
1st feature 2nd feature 3rd feature 4th feature nth feature
Parent 2
1 0 0 0 •••••••••••••••••• 1
1st feature 2nd feature 3rd feature 4th feature nth feature
uniform crossover
✓ Prob < cross_rate
✕ Prob > cross_rate
✓ ✓ ✕ ✓ ✕
Offspring 1
1 0 1 0 •••••••••••••••••• 1
1st feature 2nd feature 3rd feature 4th feature nth feature
Offspring 2
0 0 0 1 •••••••••••••••••• 1
1st feature 2nd feature 3rd feature 4th feature nth feature

Figure 3.4 – Uniform Crossover

3.2.6 Mutation

Offspring chromosomes formed after crossover between parents are further explored with the mutation operator. The idea being having combined the information from two good chromosomes with the possibility of retaining the parts that are responsible for the good-ness, it might be the case that an addition or removal of a good or bad feature or two respectively could result in a little more exploration and make it even better. The rate of mutation decides the amount of disruption to be caused. It is usually kept very low since a high mutation rate may cause the offspring to lose whatever information it gained from the combination of its parents and derail the search. For the length of a chromosome, a bit position is switched if it is less than the mutation rate otherwise left alone. Mutation is also necessary to break deadlock situations which may arise while performing crossover between two instances of the same chromosome. Such a situation is possible since a highly fit chromosome can get selected multiple times.

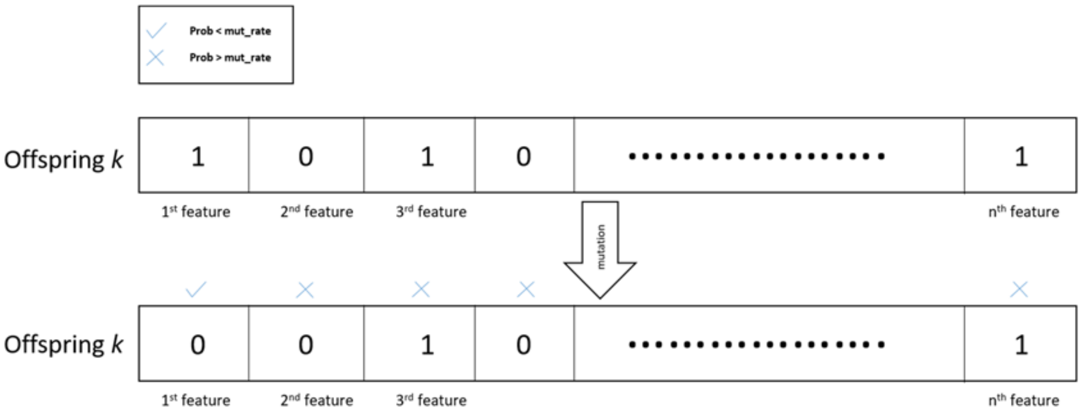

Figure 3.5 - Mutation

3.2.7 Parallelization

The past few decades have seen an increase in computational prowess in multiples. With the advent of new-age parallel architectures that leverage the additive performance of multiple processors, computational throughput has increased manifolds. Executing multiple tasks in parallel can significantly reduce the computing time of high complexity jobs, especially the ones which has discrete modules. In addition to the implicit parallelizability of a genetic algorithm (a set of solutions are investigated at a time), it is possible to reduce the time for search by modifying it into a suitable parallel implementation by employing a number of computer processes with distributed or shared memories. The concerns regarding such a parallel implementation range from the number of available processors to the inter processor communication time and computation time of the distributable tasks. The strategies that are in use can be categorized into two broad groups. The “island” model divides the entire population into a number of distinct subpopulations which are evolved independently and simultaneously with some amount of migration between them. The “farming” model uses a master-slave approach in which the population resides on a master processor and the fitness computation of the chromosomes is distributed to multiple slave processors [106].

These modules form the foundation of the genetic algorithm. The algorithm starts with the initialization of the population followed by evaluation of the fitness of the chromosomes. This forms the first generation of individuals.

Thereafter, the selection operator iteratively selects parent chromosomes, crossover is applied to mate them, the offsprings are further mutated followed by their fitness evaluation forming the population of the new generation and this process is repeatedly carried out until a stopping condition is met.

Deciding on the making of the new population and the stopping condition is an essential cog in the wheel. The algorithm is supposed to ensure the evolution of the solutions in successive generations. The new population has to be a representative of that notion. Although the inherent mechanism of the genetic algorithm is evolutionary, it can be further supported either by keeping a check on the new chromosomes formed or deciding on a replacement strategy. A popular technique is to retain the best chromosome from the previous generation if none of the chromosomes of the new population is able to perform better. Such an elitist strategy has been shown to perform well when there is sufficient exploration. Among the replacement strategies, the crowding scheme makes the new chromosome replace the worst chromosome of the previous generation if it has a better fitness [107]. Finally, determining the stopping condition is also crucial for the genetic algorithm to be able to locate a good solution. Common strategies include fixing a number of generations for which the algorithm is allowed to run, fixing a threshold fitness value or specifying a saturation point [108].

Algorithm: GA

Input:

pop_size: population size
pc: crossover rate
pm: mutation rate
g: number of generations
s: saturation

Procedure:

```
while |population|<pop_size
  {
     generate new chromosome c and evaluate fitness f[c]
  }
best=c in population with maximum f[c];saturation=0
do{
     while |new population|<pop_size
         {
             select two chromosomes from population p1 and p2
             generate random number x€[0,1]
             if x<pc
             {
                Crossover p1 and p2 forming c1 and c2
                mutate c1 and add to new population
                mutate c2 and add to new population
             }
            else
           {
              if p1 and p2 not in child population
                 add to child population
             else
                 continue
           }
         }
      worst=c in new population with minimum f[c]
      if maximum f[c] in new population <f[best]
        replace worst with best
        saturation+=1
     else if maximum f[c] in new population=f[best] and number of 1 in c with maximum f[c]>=number of 1 in best
         replace worst with best
        saturation+=1
     else
        best=c in new population with maximum f[c]
        saturation=0
} while generation<gen and saturation<s
return best
```

Output:

Best chromosome

Chapter 4

Materials and methods

This thesis has proceeded to work on the problem of predicting S-nitrosylation post-translation modification sites in proteins. For the purpose, a feature optimization technique has been proposed to improve predictive performance. It ensues by clustering of the features followed by optimization in two steps. Genetic algorithm is used as a wrapper based feature selection method to find the optimal subset of features within the clusters, the resulting subsets are then merged and optimized using another iteration of the GA. The performance of a classifier machine learning algorithm is used as the objective function for the GA. The following sections elaborate the steps taken to carry out the proposed process.

4.1 Data collection and preparation

The benchmark dataset used in this work has been downloaded from the dbPTM (http://dbPTM.mbc.nctu.edu.tw/) [68] database that contains information about multiple PTMs. The database contains non-homologous benchmark datasets for individual PTM types which are curated by the CD-HIT program [109] for homology reduction. After homology reduction, the resulting sequences are labelled by mentioning the sequence positions of the residues annotated to be modified protein-wise in the positive set and the non-modified residues of the same type in the negative set. Moreover, to reduce the imbalance between the negative and the positive set, a further homology reduction step has been performed on the negative set and the sequences that are found to be at a similarity threshold that is higher than what prevails in the positive set are removed. The extracted data for S-nitrosylation contains 3592 sites in 2077 proteins in the positive set and 5803 sites in 1434 proteins in the negative set. The corresponding protein sequences are then derived from the uniprot knowledgebase (UniprotKB) of the Uniprot (http://www.uniprot.org) [11] database. In order to reduce the imbalance some more, 4000 negative sites selected randomly are considered as negative data. Post this 283 sites in 82 proteins are separated from the dataset to be used as a test set for final performance evaluation and comparison with other S-nitrosylation predictors. The filtered dataset now contains 3458 sites in 1495 proteins as positives and 4000 sites in 1267 proteins as negatives to be considered as training instances.

4.2 Classifier model

The support vector machine (SVM) [110] is chosen as the learning algorithm to carry out the classification, it being one of the most preferred classifiers when the problem is a binary classification. With respect to PTM related predictions, it has been shown to perform well compared to other shallow classification algorithms such as Random Forest, KNN and a few others [111]. The SVM tries to create a decision hyperplane in the data space between the two classes such that it is separated from them as much as possible. It aims to achieve this by projecting the data into a higher dimension and selecting certain data points, called support vectors, which it uses to decide on the separation. This makes it better suited for high-dimensional datasets. The publicly available SVM package scikit-learn is used to design the classifier using a polynomial kernel.

4.3 Feature extraction and training sample representation

Next, the lengths of 17 to 21 are chosen as values for $\boldsymbol{m}$ to describe the reference frame of each site instance. Thereafter, the sub-sequences of interest with respect to each site are extracted from their respective protein sequences as windows of length $\boldsymbol{2m+1}$ with Cysteine at the centre. The window that provides the best performance is selected to be worked upon further. A rather crude similarity negation step is carried out while generating the subsequences by restricting any overlap between two subsequences. For the features, in the present work, 566 different feature descriptors viz,. various physicochemical and biochemical indices from the AAindex (http://www.genome.ad.jp/aaindex/) [90] database have been used to describe an instance initially. Also, in order to retain some of the sequence positional information of the amino acid residues, a feature called the Position Specific Amino Acid Propensity (PSAAP) have been adopted from [112]. The PSAAP is defined as a matrix of dimension $\mathbf{20} \times \mathbf{2}\boldsymbol{m} + \mathbf{1}$ where an entry in the ith row and the jth column indicates the propensity of the ith amino acid in the set of 20 amino acids arranged alphabetically to occur at the jth position in the subsequence. The PSAAP takes into account the negative imbalance by averaging over $\boldsymbol{b}$ bins of size $\boldsymbol{P}$ each, where $\boldsymbol{b} = \boldsymbol{P}/\boldsymbol{N}$, if $\boldsymbol{N}$ and $\boldsymbol{P}$ are the total number of negative and positive sequences respectively. Then an entry in the i, j-th position of the PSAAP matrix is be given by

$$PSSAP_{i,j} = \frac{f(i,j)_{\boldsymbol{P}} - \overline{f(i,j)}_{\forall \boldsymbol{b}}}{\sigma_{f(i,j)_{\forall \boldsymbol{b}}}} \tag{10}$$

Where $f(i,j)$ is the occurrence frequency of amino acid i at position j.

A feature then comprises of a $(2m + 1) + (2m + 1)$ length max-normalized vector for each sub-sequence, formed by substituting the amino acids with their values of the respective property from the AAindex and their propensity value from the PSAAP.

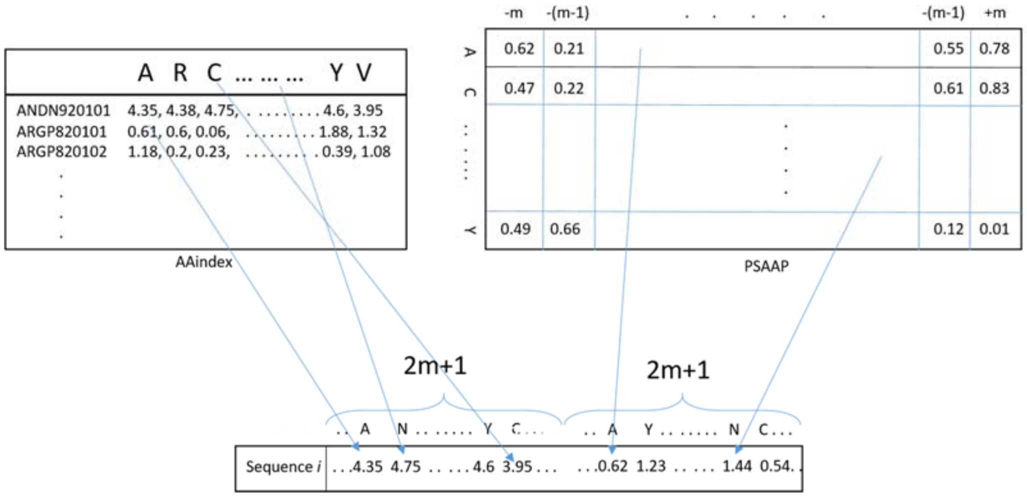
A R C … … … Y V
ANDN920101 4.35, 4.38, 4.75, ... 4.6, 3.95
ARGP820101 0.61, 0.6, 0.06, ... 1.88, 1.32
ARGP820102 1.18, 0.2, 0.23, ... 0.39, 1.08
AAindex
-m -(m-1) . . . . . . -(m-1) +m
A 0.62 0.21 0.55 0.78
C 0.47 0.22 0.61 0.83
Y 0.49 0.66 0.12 0.01
PSAAP
2m+1
2m+1
. . A N . . . . . . Y C . . .
. . A Y . . . . . . N C . . .
Sequence i . . .4.35 4.75 . . . . .4.6 3.95 . . . . . .0.62 1.23 . . . . . 1.44 0.54. .

Figure 4.1 – An example of an instance vector

4.4 Feature optimization

A multitude of steps is performed next to optimize the set of features. The 566 indices from the AAindex are first clustered using the method described in [113] which uses a hierarchical clustering technique and a silhouette score as a measure for the validity of a cluster. This is done to group together the features which are similar based on their descripting values. This returns 331 non-singleton and 185 singleton clusters. A genetic algorithm is then used to optimize the features corresponding to the clustered AAindices in two steps. A first run of the GA selects the best performing features from within each cluster. The so obtained features are then merged together and the GA is run a second time on these features. The GA is parallelized based on the master-slave model. The features indicated by each chromosome is used to form the training data for an SVM model. The ratio of the positive and negative samples is kept at 1:1 for which the negative instances are randomly sampled from the total set of negative instances. The model performance is averaged over 3 equal folds of cross-validation, each on a different processor. MCC is selected as the evaluation metric which is used as the objective function by the genetic algorithm. The implementation of the genetic algorithm designed includes a Roulette wheel as the selection operator. The crossover between two parents is performed uniformly across the length of the chromosome and its subsequent mutation. Existence of duplicate solutions in the population is checked and weeded out. The chromosomes forming the new population is checked for twins, if found they are discarded and the creation of another chromosome takes place. This is done to ensure diversity in the population. Along with this, the best performing individual is retained in the succeeding generation if a better solution is not found. To break a tie between two equally performing chromosomes, the one with the lower number features is retained. Finally, the global best chromosome is returned. A successful iteration of the GA results in a subset of the original features. The GA is restarted on this feature subset. This continues until a recursive instance is able to improve the performance further.

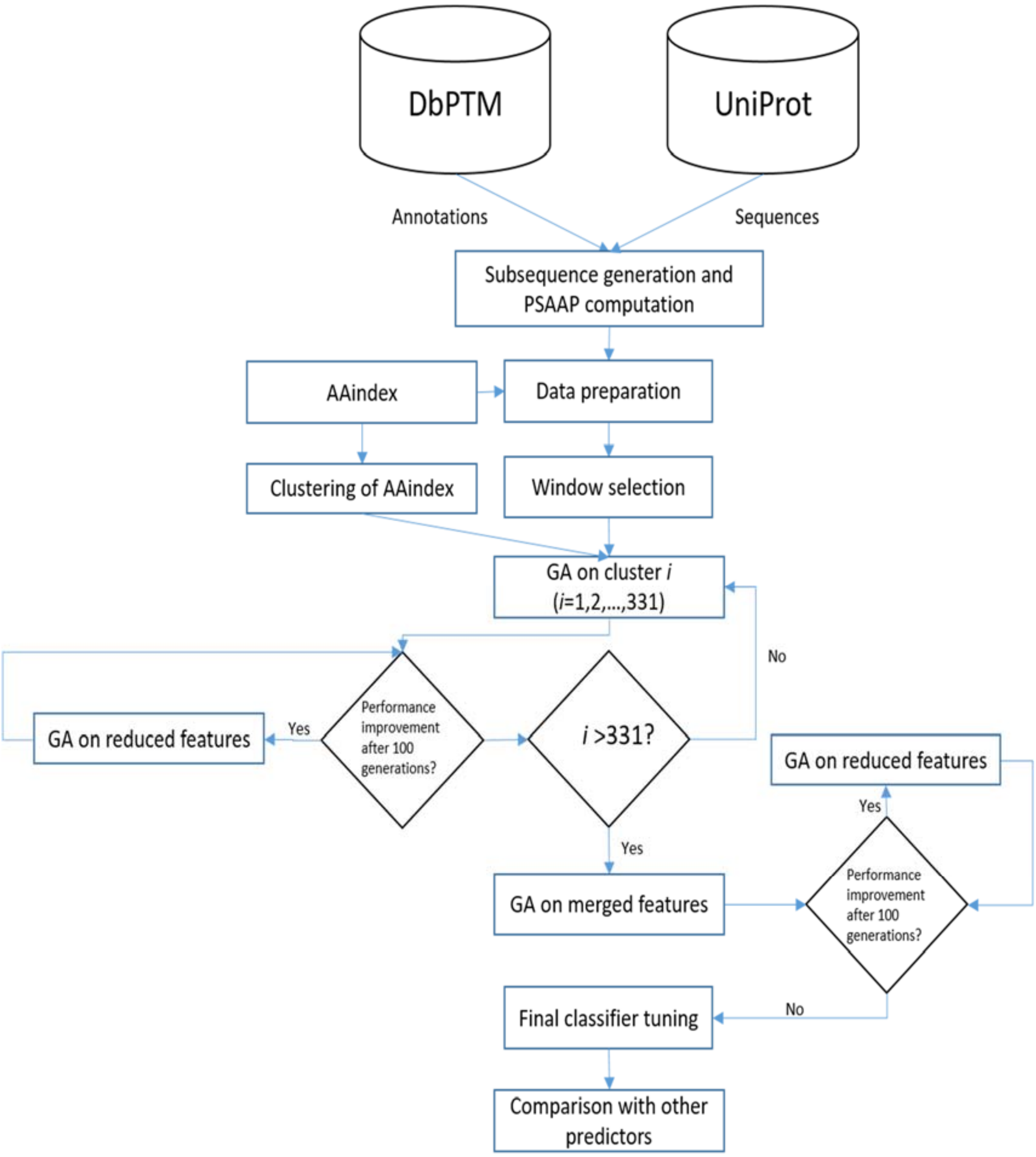
DbPTM
UniProt
Annotations
Sequences
Subsequence generation and PSAAP computation
AAindex
Data preparation
Clustering of AAindex
Window selection
GA on cluster *i* (*i*=1,2,...,331)
No
GA on reduced features
Yes
Performance improvement after 100 generations?
i >331?
GA on reduced features
Yes
Yes
GA on merged features
Performance improvement after 100 generations?
Final classifier tuning
No
Comparison with other predictors

Figure 4.2 - Method flowchart

4.5 Parameter Selection

A number of parameters have been used across the designing of the method. The SVM accepts two essential parameters. The kernel parameter '*gamma'* which is set to 0.02 decides the radius of influence of samples from the decision plane. The penalty parameter *'C'* indicates the stringency of the classifier and how it treats wrongly classified instances. This is set to 1 which is otherwise also its default value. The essential parameters for the genetic algorithm are population size, crossover rate, mutation rate, and the stopping condition. The population size is set to 15. This decision is primarily based on the number of available processors. The crossover rate is set at 0.7 and the mutation rate at 0.01. 100 generations are fixed as the stopping condition along with a saturation criteria. If the global best fitness does not improve over 50 consecutive generations the iteration is terminated.

Chapter 5

Experimental Results

The experimental results obtained in multiple stages of the method are described in this chapter.

Table 5.1 shows the performance of the classifier on different window lengths using all the features together. The length of 19 was found to give the best result and was subsequently selected for further experimentation.

Table 5.1 – Performance of different window lengths

	17	18	19	20	21
F1	0.707	0.710	0.714	0.712	0.708
AUC	0.783	0.785	0.788	0.785	0.784
MCC	0.384	0.390	0.411	0.4	0.387

Having fixated on the window the features were tested individually and ranked according to their MCC scores. Table 5.2 shows the top 20 performing indices.

Table 5.2 – Top 20 performing features ranked by MCC

Features named by AAindex	F1	AUC	MCC
BIOV880101	0.688	0.767	0.397
RICJ880108	0.703	0.767	0.395
OOBM850103	0.675	0.764	0.395
MEEJ810102	0.717	0.768	0.394
MEIH800103	0.668	0.766	0.393
MEIH800102	0.699	0.766	0.392
HUTJ700102	0.718	0.768	0.392
MIYS850101	0.71	0.768	0.392
PALJ810104	0.702	0.764	0.391
WARP780101	0.7	0.767	0.389
ROSG850102	0.678	0.767	0.389
NADH010102	0.685	0.769	0.389
MANP780101	0.662	0.765	0.389
OOBM770103	0.677	0.766	0.388
TSAJ990102	0.653	0.762	0.388
ZHOH040103	0.681	0.765	0.388
GEIM800105	0.705	0.763	0.388
KARS160106	0.658	0.76	0.388
BIGC670101	0.669	0.762	0.387
RADA880108	0.697	0.762	0.387

Following this, forward selection was performed on the ranked features. Table 5.3 shows the indices that were selected as the result of forward selection.

Table 5.3 – Features selected by forward selection

Features named by AAindex	F1	AUC	MCC
BIOV880101 RICJ880108 OOBM850103 MEEJ810102 MEIH800103 MEIH800102 HUTJ700102	0.721	0.770	0.399

Next, the indices were clustered using their raw values from the AAindex. This resulted in 331 clusters, of which 185 were singleton clusters. The genetic algorithm was run on each individual cluster first. 441 indices were selected from the 566 initially over all the clusters. The GA was then executed again on these 441 indices which finally resulted in the 55 indices.

Table 5.4 – Largest 10 clusters of feature descriptors formed and their reductions after two stages of GA

CLUSTER NUMBER	INITIAL	AFTER INTRA-CLUSTER GA	AFTER INTER-CLUSTER GA
Cluster16	PALJ810101 LEVM780104 GEIM800101 PRAM900102 LEVM780101 PALJ810102 CHOP780201 MAXF760101 KANM800101 ISOY800101 ROBB760101 CRAJ730101 BURA740101 TANS770101 NAGK730101 PALJ810109 GEIM800104	PALJ810101 CRAJ730101 BURA740101	KANM800103 RACS820108 ISOY800101 CRAJ730101 RICJ880112 PALJ810108 HUTJ700103 HUTJ700102 QIAN880112 FINA910104 KARS160111 DAWD720101 JACR890101 FASG760105 NADH010103 CORJ870101 TANS770103 NAGK730102 CHOP780209 PALJ810112 PTIO830102 WIMW960101 ZASB820101 VASM830103 RICJ880111 COSI940101 CHOP780203 CHOP780211 PARS000102 KARP850103 FAUJ880110 JANJ780101 ISOY800107 RICJ880113 MEIH800102 KARP850101 BULH740101 NAKH920102 JUKT750101 NAKH920107 FUKS010105 GEOR030104 AURR980119 FASG760104 ISOY800106 WEBA780101 JOND920102 OOBM770105 OOBM850104 PALJ810114 CHAM820102 WILM950103 WILM950104 GEIM800103 ROBB760111
Cluster38	GOLD730102 BIGC670101 KRIW790103 GRAR740103 TSAJ990102 TSAJ990101 CHOC750101 PONJ960101 HARY940101	PONJ960101	
Cluster51	ROBB760103 PTIO830101 ROBB760104 QIAN880109 QIAN880108 QIAN880110	ROBB760103 QIAN880108 QIAN880110	
Cluster56	FAUJ880113 FASG760103 BLAM930101 ONEK900101 BUNA790101	ONEK900101	
Cluster116	PONP930101 MANP780101 PONP800108 NISK800101 CORJ870101 PONP800102 PONP800101 PONP800103	PONP800103	
Cluster156	SIMZ760101 GOLD730101 JOND750101 ARGP820101	ARGP820101	
Cluster183	FAUJ880108 CHOP780212 RACS820104 GRAR740101	RACS820104 GRAR740101	
Cluster233	KIDA850101 ROSM880102 KUHL950101 ROSM880101	ROSM880102 KUHL950101	
Cluster235	GUYH850104 JANJ780101 JANJ780103 CHOC760102	GUYH850104 JANJ780101	

Table 5.5 – Performance comparison of feature selection

	All features	Forward Selection	GA
F1	0.714	0.721	0.719
AUC	0.788	0.770	0.778
MCC	0.411	0.399	0.468

It can be observed from the results (Table 5.5) that the genetic algorithm was able to converge on a subset of features that improved its performance as compared to using all available features. To compare the predictor, it was then tested with three other S-nitrosylation predictors GPS-SNO [114], SNOSite [72] and DeepNitro [115] that had a web-server running or an executable available that could be downloaded. Further, the GPS-SNO allowed testing at the decision thresholds of low, medium and high. As is the practice that a PTM predictor should accept entire protein sequences, the sequences of the 82 proteins separated into the test set at the beginning of experimentation containing 283 cysteine sites were submitted to all the predictors. The performance scores thus obtained are given in Table 5.6.

Table 5.6 – Comparison of designed predictor with other predictors from the literature

	GPS_SNO (low)	GPS_SNO (medium)	GPS_SNO (high)	SNOSite	DeepNitro	Proposed Method
Accuracy	0.593	0.611	0.636	0.636	0.72	0.625
F1	0.444	0.395	0.343	0.611	0.582	0.646
MCC	0.127	0.136	0.175	0.299	0.395	0.351

Chapter 6

Conclusion

Even several years after the publication of the human genome, the largest functional category of the predicted and known genes, is the one labelled “function unknown”. Classification of these proteins which have no homology to known proteins represents a gigantic experimental task. Prediction methods may aid in solving this task. For example, it has been shown that some proteins, which are related functionally, but not related at the sequence or structure levels, share some of the same PTM features. PTMs are therefore significant biomarkers and of interest much beyond the individual sequence, to understanding evolutionary pressures that go beyond maintaining protein structure. It might be speculated, that for some proteins the ability to become modified is more important than to preserve its three-dimensional structure. It is not unlikely, that the understanding of protein function in the coming years will involve PTMs in a much more prominent role, and in that sense balance the picture which so far has mostly been based on protein structure. The thesis is intended at designing a robust and reliable predictor for potential S-nitrosylation sites in proteins which is also able to provide an insight into the analysis of biological relations that are responsible for the event. Results obtained indicate, that the predictor is able to achieve comparable performance with other S-nitrosylation predictors. This work proposes a feature optimization technique on two levels. As a preprocessing step, the feature descriptors used to formulate the problem space, are clustered to group the ones together that are similar. A recursive implementation of a GA is then used to optimize the features intra-cluster and inter-cluster. A much regrettable fact is that, though machine learning techniques are able to model complex correlated factors and are hence highly sought for, they are completely mathematically oriented and the models that they generate are extremely hard to interpret back to human readable logic. The genetic algorithm based optimization of attributes proposed in this work can alleviate the issue by providing a way to understand what attributes are at work, which can be accepted with a reasonable compromise. Along with the analysis of predicted PTM sites, the analysis of the selected features and the ones excluded should reveal important hints to the concerned. The work reveals a few threads that can be improved upon in the future. An issue with the PSAAP is that it is not able to factor correlations between residues. Currently, the physicochemical properties of the surrounding environment of the modification site were used as the main structural information to develop the predictor. However, the enzymes involved in reactions exhibit specificity not only for the amino acid residues to be modified but also for the tertiary structure of protein substrates. Studies have indicated that a distinct subset of reactions occurs at residues buried deeply in the proteins. Therefore structure-based properties need to be compulsorily integrated. There exist a number of other properties of a protein which are possible major players in its function. Cellular localization and the interaction pathways to name a few. All of these pose as important candidates for features. The genetic algorithm also has multiple scope for improvement. A purely stochastic GA has to deal with an issue of premature convergence. Local gradient-based search added to it as support could improve its performance. Finally, the proposed method as a feature optimization technique for PTMs can be generalized to other PTMs.

References

[1] W. T. Astbury, “Molecular Biology or Ultrastructural Biology ?,” *Nature*, vol. 190, no. 4781, pp. 1124–1124, Jun. 1961.

[2] A. Uzman *et al.*, “Molecular biology of the cell (4th ed.),” *Biochem. Mol. Biol. Educ.*, vol. 31, no. 4, pp. 212–214, Jul. 2003.

[3] H. F. Lodish, D. Baltimore, A. Berk, S. Lawrence Zipursky, P. Matsudaira, and J. E Darnell, *In Molecular Cell Biology*, vol. 4. 1995.

[4] P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan, “Predicting function: from genes to genomes and back 1 1Edited by P. E. Wright,” *J. Mol. Biol.*, vol. 283, no. 4, pp. 707–725, Nov. 1998.

[5] C. Walsh, *Posttranslational modification of proteins : expanding nature’s inventory*. Roberts and Co. Publishers, 2006.

[6] M. Uhlen and F. Ponten, “Antibody-based proteomics for human tissue profiling.,” *Mol. Cell. Proteomics*, vol. 4, no. 4, pp. 384–93, Apr. 2005.

[7] O. Nørregaard Jensen, “Modification-specific proteomics: characterization of post-translational modifications by mass spectrometry,” *Curr. Opin. Chem. Biol.*, vol. 8, no. 1, pp. 33–41, Feb. 2004.

[8] W. N. Burnette, “‘Western Blotting’: Electrophoretic transfer of proteins from sodium dodecyl sulfate-polyacrylamide gels to unmodified nitrocellulose and radiographic detection with antibody and radioiodinated protein A,” *Anal. Biochem.*, vol. 112, no. 2, pp. 195–203, Apr. 1981.

[9] O. D. (Orrin D. Sparkman, *Mass spectrometry desk reference*. Global View Pub, 2000.

[10] H. Li *et al.*, “SysPTM: A Systematic Resource for Proteomic Research on Post-translational Modifications,” *Mol. Cell. Proteomics*, vol. 8, no. 8, pp. 1839–1849, Aug. 2009.

[11] The UniProt Consortium, “UniProt: the universal protein knowledgebase.,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D158–D169, 2017.

[12] D. L. (David L. Nelson, M. M. Cox, and A. L. Lehninger, *Principles of biochemistry*. W. H. Freeman, 2017.

[13] F. Crick, “Central Dogma of Molecular Biology,” *Nature*, vol. 227, no. 5258, pp. 561–563, Aug. 1970.

[14] C. Chung, “Machine Learning Approaches to Refining Post-translational Modification Predictions and Protein Identifications from Tandem Mass Spectrometry,” 2012.

[15] K. S. Kolibaba and B. J. Druker, “Protein tyrosine kinases and cancer.,” *Biochim. Biophys. Acta*, vol. 1333, no. 3, pp. F217-48, Dec. 1997.

[16] L. N. Johnson, M. E. . Noble, and D. J. Owen, “Active and Inactive Protein Kinases: Structural Basis for Regulation,” *Cell*, vol. 85, no. 2, pp. 149–158, Apr. 1996.

[17] K. L. Agarwal, G. W. Kenner, and R. C. Sheppard, “Feline gastrin. An example of peptide sequence analysis by mass spectrometry.,” *J. Am. Chem. Soc.*, vol. 91, no. 11, pp. 3096–7, May 1969.

[18] W. Ni, “Advances in protein post-translational modifications (PTMS) using liquid chromatography-mass spectrometry.” 2013.

[19] M. Jezek, A. Jacques, D. Jaiswal, and E. M. Green, “Chromatin Immunoprecipitation (ChIP) of Histone Modifications from &lt;em&gt;Saccharomyces cerevisiae&lt;/em&gt;,” *J. Vis. Exp.*, no. 130, Dec. 2017.

[20] D. J. Slade, V. Subramanian, J. Fuhrmann, and P. R. Thompson, “Chemical and biological methods to detect post-translational modifications of arginine,” *Biopolymers*, vol. 101, no. 2, pp. 133–143, Feb. 2014.

[21] Y. V Karpievitch, A. D. Polpitiya, G. A. Anderson, R. D. Smith, and A. R. Dabney, “Liquid Chromatography Mass Spectrometry-Based Proteomics: Biological and Technological Aspects.,” *Ann. Appl. Stat.*, vol. 4, no. 4, pp. 1797–1823, 2010.

[22] J. Eswaran and S. Knapp, “Insights into protein kinase regulation and inhibition by large scale structural comparison.,” *Biochim. Biophys. Acta*, vol. 1804, no. 3, pp. 429–32, Mar. 2010.

[23] G. Duan and D. Walther, “The roles of post-translational modifications in the context of protein interaction networks.,” *PLoS Comput. Biol.*, vol. 11, no. 2, p. e1004049, Feb. 2015.

[24] D. L. Swaney *et al.*, “Global analysis of phosphorylation and ubiquitylation cross-talk in protein degradation,” *Nat. Methods*, vol. 10, no. 7, pp. 676–682, Jul. 2013.

[25] C. Alabert *et al.*, “Two distinct modes for propagation of histone PTMs across the cell cycle.,” *Genes Dev.*, vol. 29, no. 6, pp. 585–90, Mar. 2015.

[26] P. J. Hurd *et al.*, “Phosphorylation of histone H3 Thr-45 is linked to apoptosis.,” *J. Biol. Chem.*, vol. 284, no. 24, pp. 16575–83, Jun. 2009.

[27] C. M. Spickett, A. R. Pitt, N. Morrice, and W. Kolch, “Proteomic analysis of phosphorylation, oxidation and nitrosylation in signal transduction,” *Biochim. Biophys. Acta - Proteins Proteomics*, vol. 1764, no. 12, pp. 1823–1841, Dec. 2006.

[28] S. LI, L. M. IAKOUCHEVA, S. D. MOONEY, and P. RADIVOJAC, “LOSS OF POST-TRANSLATIONAL MODIFICATION SITES IN DISEASE,” in *Biocomputing 2010*, WORLD SCIENTIFIC, 2009, pp. 337–347.

[29] G. H. Eom and H. Kook, “Posttranslational modifications of histone deacetylases: Implications for cardiovascular diseases,” *Pharmacol. Ther.*, vol. 143, no. 2, pp. 168–180, Aug. 2014.

[30] G. Harauz, N. Ishiyama, C. M. . Hill, I. R. Bates, D. S. Libich, and C. Farès, “Myelin basic protein—diverse conformational states of an intrinsically unstructured protein and its roles in myelin assembly and multiple sclerosis,” *Micron*, vol. 35, no. 7, pp. 503–542, Oct. 2004.

[31] A. M. Bode and Z. Dong, “Post-translational modification of p53 in tumorigenesis,” *Nat. Rev. Cancer*, vol. 4, no. 10, pp. 793–805, Oct. 2004.

[32] J. Reimand, O. Wagih, and G. D. Bader, “Evolutionary Constraint and Disease Associations of Post-Translational Modification Sites in Human Genomes,” *PLOS Genet.*, vol. 11, no. 1, p. e1004919, Jan. 2015.

[33] P. Radivojac, P. H. Baenziger, M. G. Kann, M. E. Mort, M. W. Hahn, and S. D. Mooney, “Gain and loss of phosphorylation sites in human cancer,” *Bioinformatics*, vol. 24, no. 16, pp. i241–i247, Aug. 2008.

[34] T. Kitada *et al.*, “Mutations in the parkin gene cause autosomal recessive juvenile parkinsonism,” *Nature*, vol. 392, no. 6676, pp. 605–608, Apr. 1998.

[35] J.-Z. Wang, I. Grundke-Iqbal, and K. Iqbal, “Glycosylation of microtubule–associated protein tau: An abnormal posttranslational modification in Alzheimer’s disease,” *Nat. Med.*, vol. 2, no. 8, pp. 871–875, Aug. 1996.

[36] A. Sharma *et al.*, “Genetic association, post-translational modification, and protein-protein interactions in Type 2 diabetes mellitus.,” *Mol. Cell. Proteomics*, vol. 4, no. 8, pp. 1029–37, Aug. 2005.

[37] D. T. Hess, A. Matsumoto, S.-O. Kim, H. E. Marshall, and J. S. Stamler, “Protein S-nitrosylation: purview and parameters,” *Nat. Rev. Mol. Cell Biol.*, vol. 6, no. 2, pp. 150–166, Feb. 2005.

[38] J. S. Stamler, “Redox signaling: nitrosylation and related target interactions of nitric oxide.,” *Cell*, vol. 78, no. 6, pp. 931–6, Sep. 1994.

[39] J. S. Stamler, S. Lamas, and F. C. Fang, “Nitrosylation: The Prototypic Redox-Based Signaling Mechanism,” *Cell*, vol. 106, no. 6, pp. 675–683, Sep. 2001.

[40] B. M. Gaston, J. Carver, A. Doctor, and L. A. Palmer, “S-nitrosylation signaling in cell biology.,” *Mol. Interv.*, vol. 3, no. 5, pp. 253–63, Aug. 2003.

[41] M. Benhar, M. T. Forrester, and J. S. Stamler, “Protein denitrosylation: enzymatic mechanisms and cellular functions,” *Nat. Rev. Mol. Cell Biol.*, vol. 10, no. 10, pp. 721–732, Oct. 2009.

[42] D. T. Hess, A. Matsumoto, R. Nudelman, and J. S. Stamler, “S-nitrosylation: spectrum and specificity,” *Nat. Cell Biol.*, vol. 3, no. 2, pp. E46–E48, Feb. 2001.

[43] F. Li *et al.*, “Regulation of HIF-1α Stability through S-Nitrosylation,” *Mol. Cell*, vol. 26, no. 1, pp. 63–74, Apr. 2007.

[44] E. J. Whalen *et al.*, “Regulation of β-Adrenergic Receptor Signaling by S-Nitrosylation of G-Protein-Coupled Receptor Kinase 2,” *Cell*, vol. 129, no. 3, pp. 511–522, May 2007.

[45] A. A. Lugovskoy, P. Zhou, J. J. Chou, J. S. McCarty, P. Li, and G. Wagner, “Solution Structure of the CIDE-N Domain of CIDE-B and a Model for CIDE-N/CIDE-N Interactions in the DNA Fragmentation Pathway of Apoptosis,” *Cell*, vol. 99, no. 7, pp. 747–755, Dec. 1999.

[46] S. Korde Choudhari, M. Chaudhary, S. Bagde, A. R. Gadbail, and V. Joshi, “Nitric oxide and cancer: a review,” *World J. Surg. Oncol.*, vol. 11, no. 1, p. 118, Dec. 2013.

[47] C. M. Schonhoff *et al.*, “S-nitrosothiol depletion in amyotrophic lateral sclerosis.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 7, pp. 2404–9, Feb. 2006.

[48] M. W. Akhtar, C. R. Sunico, T. Nakamura, and S. A. Lipton, “Redox Regulation of Protein Function via Cysteine S-Nitrosylation and Its Relevance to Neurodegenerative Diseases,” *Int. J. Cell Biol.*, vol. 2012, pp. 1–9, Aug. 2012.

[49] M. W. Foster, D. T. Hess, and J. S. Stamler, “Protein S-nitrosylation in health and disease: a current perspective,” *Trends Mol. Med.*, vol. 15, no. 9, pp. 391–404, Sep. 2009.

[50] D. T. Hess and J. S. Stamler, “Regulation by S-nitrosylation of protein post-translational modification.,” *J. Biol. Chem.*, vol. 287, no. 7, pp. 4411–8, Feb. 2012.

[51] J. Astier *et al.*, *S-nitrosylation: An emerging post-translational protein modification in plants*, vol. 181. 2011.

[52] S. R. Jaffrey and S. H. Snyder, “The biotin switch method for the detection of S-nitrosylated proteins.,” *Sci. STKE*, vol. 2001, no. 86, p. pl1, Jun. 2001.

[53] C. Chen, H. Huang, and C. H. Wu, “Protein Bioinformatics Databases and Resources.,” *Methods Mol. Biol.*, vol. 1558, pp. 3–39, 2017.

[54] R. J. Singh, N. Hogg, J. Joseph, and B. Kalyanaraman, "Mechanism of nitric oxide release from S-nitrosothiols.," *J. Biol. Chem.*, vol. 271, no. 31, pp. 18596–603, Aug. 1996.

[55] J. N. Smith and T. P. Dasgupta, "Kinetics and Mechanism of the Decomposition of S-Nitrosoglutathione by l-Ascorbic Acid and Copper Ions in Aqueous Solution to Produce Nitric Oxide," *Nitric Oxide*, vol. 4, no. 1, pp. 57–66, Feb. 2000.

[56] T. Madden, "The BLAST Sequence Analysis Tool," Mar. 2013.

[57] L. A. Pinna and M. Ruzzene, "How do protein kinases recognize their substrates?," *Biochim. Biophys. Acta - Mol. Cell Res.*, vol. 1314, no. 3, pp. 191–225, Dec. 1996.

[58] R. Linding *et al.*, "Systematic Discovery of In Vivo Phosphorylation Networks," *Cell*, vol. 129, no. 7, pp. 1415–1426, Jun. 2007.

[59] R. M. Biondi and A. R. Nebreda, "Signalling specificity of Ser/Thr protein kinases through docking-site-mediated interactions.," *Biochem. J.*, vol. 372, no. Pt 1, pp. 1–13, May 2003.

[60] P. M. Holland and J. A. Cooper, "Protein modification: Docking sites for kinases," *Curr. Biol.*, vol. 9, no. 9, pp. R329–R331, May 1999.

[61] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Res.*, vol. 22, no. 22, pp. 4673–4680, Nov. 1994.

[62] C. M. Bishop, "Pattern Recognition and Machine Learning Springer Mathematical notation Ni."

[63] M. Audagnotto and M. Dal Peraro, "Protein post-translational modifications: In silico prediction tools and molecular modeling," *Comput. Struct. Biotechnol. J.*, vol. 15, pp. 307–319, Jan. 2017.

[64] D. Plewczynski, A. Tkacz, L. S. Wyrwicz, and L. Rychlewski, "AutoMotif server: Prediction of single residue post-translational modifications in proteins,"

Bioinformatics, vol. 21, no. 10, pp. 2525–2527, 2005.

[65] X. Chen, S. P. Shi, H. D. Xu, S. B. Suo, and J. D. Qiu, “A homology-based pipeline for global prediction of post-translational modification sites,” *Sci. Rep.*, vol. 6, no. May, pp. 1–8, 2016.

[66] P. V. Hornbeck, I. Chabra, J. M. Kornhauser, E. Skrzypek, and B. Zhang, “PhosphoSite: A bioinformatics resource dedicated to physiological protein phosphorylation,” *Proteomics*, vol. 4, no. 6, pp. 1551–1561, Jun. 2004.

[67] R. Gupta, H. Birch, K. Rapacki, S. Brunak, and J. E. Hansen, “O-GLYCBASE version 4.0: a revised database of O-glycosylated proteins,” *Nucleic Acids Res.*, vol. 27, no. 1, pp. 370–372, Jan. 1999.

[68] K. Huang *et al.*, “dbPTM in 2019 : exploring disease association and cross-talk of post-translational modifications,” vol. 47, no. November 2018, pp. 298–308, 2019.

[69] S. Basu and D. Plewczynski, “AMS 3.0: prediction of post-translational modifications,” *BMC Bioinformatics*, vol. 11, no. 1, p. 210, Dec. 2010.

[70] Y. Xue, J. Ren, X. Gao, C. Jin, L. Wen, and X. Yao, “GPS 2.0, a tool to predict kinase-specific phosphorylation sites in hierarchy.,” *Mol. Cell. Proteomics*, vol. 7, no. 9, pp. 1598–608, Sep. 2008.

[71] J. Ren, L. Wen, X. Gao, C. Jin, Y. Xue, and X. Yao, “CSS-Palm 2.0: an updated software for palmitoylation sites prediction,” *Protein Eng. Des. Sel.*, vol. 21, no. 11, pp. 639–644, Aug. 2008.

[72] T. Lee, Y. Chen, T. Lu, H. Huang, and Y. Chen, “SNOSite : Exploiting Maximal Dependence Decomposition to Identify Cysteine S-Nitrosylation with Substrate Site Specificity,” vol. 6, no. 7, 2011.

[73] B. Wang, M. Wang, and A. Li, “Prediction of post-translational modification sites using multiple kernel support vector machine,” 2017.

[74] N. Blom, S. Gammeltoft, and S. Brunak, “Sequence and structure-based prediction of eukaryotic protein phosphorylation sites,” *J. Mol. Biol.*, vol. 294, no.

5, pp. 1351–1362, Dec. 1999.

[75] M. Lo Monte, C. Manelfi, M. Gemei, D. Corda, and A. R. Beccari, “ADPredict: ADP-ribosylation site prediction based on physicochemical and structural descriptors,” *Bioinformatics*, vol. 34, no. 15, pp. 2566–2574, 2018.

[76] J. S. Stamler, E. J. Toone, and S. A. Lipton, “(S)NO Signals: Translocation, Viewpoint Regulation, and a Consensus Motif,” 1997.

[77] V. Neduva and R. B. Russell, “Linear motifs: Evolutionary interaction switches,” *FEBS Lett.*, vol. 579, no. 15, pp. 3342–3345, Jun. 2005.

[78] P. Puntervoll *et al.*, “ELM server: a new resource for investigating short functional sites in modular eukaryotic proteins,” *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3625–3630, Jul. 2003.

[79] P. Creixell and R. Linding, “Cells, shared memory and breaking the PTM code,” *Mol. Syst. Biol.*, vol. 8, no. 1, p. 598, Jan. 2012.

[80] N. Blom, T. Sicheritz-Pontén, R. Gupta, S. Gammeltoft, and S. Brunak, “Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence,” *Proteomics*, vol. 4, no. 6, pp. 1633–1649, 2004.

[81] F. Diella *et al.*, “Understanding eukaryotic linear motifs and their role in cell signaling and regulation,” 2008.

[82] C. Pang, E. Gasteiger, and M. R. Wilkins, “Identification of arginine- and lysine-methylation in the proteome of Saccharomyces cerevisiae and its functional implications,” *BMC Genomics*, vol. 11, no. 1, p. 92, Feb. 2010.

[83] P. J. Britto, L. Knipling, and J. Wolff, “The local electrostatic environment determines cysteine reactivity of tubulin.,” *J. Biol. Chem.*, vol. 277, no. 32, pp. 29018–27, Aug. 2002.

[84] T. M. Greco *et al.*, “Identification of S-nitrosylation motifs by site-specific mapping of the S-nitrosocysteine proteome in human vascular smooth muscle cells.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 19, pp. 7420–5, May 2006.

[85] J. Jia, Z. Liu, X. Xiao, B. Liu, and K. C. Chou, “pSuc-Lys: Predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach,” *J. Theor. Biol.*, vol. 394, pp. 223–230, 2016.

[86] M. Zamani and S. Kremer, *Amino acid encoding schemes for machine learning methods*. 2011.

[87] K. Hiller, A. Grote, M. Scheer, R. Munch, and D. Jahn, “PrediSi: prediction of signal peptides and their cleavage positions,” *Nucleic Acids Res.*, vol. 32, no. Web Server, pp. W375–W379, Jul. 2004.

[88] K. C. Chou, “Using pair-coupled amino acid composition to predict protein secondary structure content.,” *J. Protein Chem.*, vol. 18, no. 4, pp. 473–80, May 1999.

[89] S. MAETSCHKE, M. TOWSEY, and M. BODÉN, “BLOMAP: AN ENCODING OF AMINO ACIDS WHICH IMPROVES SIGNAL PEPTIDE CLEAVAGE SITE PREDICTION,” in *Proceedings of the 3rd Asia-Pacific Bioinformatics Conference*, 2005, pp. 141–150.

[90] S. Kawashima and M. Kanehisa, “AAindex: Amino Acid index database,” *Nucleic Acids Res.*, vol. 28, no. 1, pp. 374–374, Jan. 2000.

[91] C. CHOTHIA, “Hydrophobic bonding and accessible surface area in proteins,” *Nature*, vol. 248, no. 5446, pp. 338–339, Mar. 1974.

[92] J. M. Sutter and J. H. Kalivas, “Comparison of Forward Selection, Backward Elimination, and Generalized Simulated Annealing for Variable Selection,” *Microchem. J.*, vol. 47, no. 1–2, pp. 60–66, Feb. 1993.

[93] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning,” *Mach. Learn.*, vol. 3, no. 2/3, pp. 95–99, 1988.

[94] S. Malakar, M. Ghosh, S. Bhowmik, R. Sarkar, and M. Nasipuri, “A GA based hierarchical feature selection approach for handwritten word recognition,” *Neural Comput. Appl.*, 2019.

[95] M. Ghosh, S. Begum, R. Sarkar, D. Chakraborty, and U. Maulik, “Recursive

Memetic Algorithm for gene selection in microarray data," *Expert Syst. Appl.*, vol. 116, pp. 172–185, 2019.

[96] M. Ghosh, S. Adhikary, K. K. Ghosh, A. Sardar, S. Begum, and R. Sarkar, "Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods," *Med. Biol. Eng. Comput.*, vol. 57, no. 1, pp. 159–176, 2019.

[97] I. Fister, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization," Jul. 2013.

[98] V. Kumar and S. Minz, "Smart Computing Review Feature Selection: A literature Review," *Smart Comput. Rev.*, vol. 4, no. 3, 2014.

[99] C. Charpilloz, A. L. Veuthey, B. Chopard, and J. L. Falcone, "Motifs tree: A new method for predicting post-translational modifications," *Bioinformatics*, vol. 30, no. 14, pp. 1974–1982, 2014.

[100] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta - Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.

[101] J. H. (John H. Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, 1992.

[102] D. W. Zingg, -M Nemec, and -T H Pulliam, "A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization."

[103] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–33, Jun. 2013.

[104] F. Sadjadi, "Comparison of fitness scaling functions in genetic algorithms with applications to optical processing," *Opt. Inf. Syst. II*, vol. 5557, p. 356, 2004.

[105] W. M. Spears and K. A. De Jong, "On the virtues of parameterized uniform crossover," *Fourth Int. Conf. Genet. Algorithms*, pp. 230--236, 1991.

[106] V. S. Gordon, D. Whitley, and S. Forrest, "Serial and Parallel Genetic Algorithms as Function Optimizers," Morgan-Kaufmann, 1993.

[107] J. A. Vasconcelos, J. A. Ramirez, R. H. C. Takahashi, and R. R. Saldanha, "Improvements in genetic algorithms," *IEEE Trans. Magn.*, vol. 37, no. 5, pp. 3414–3417, 2001.

[108] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, "On Stopping Criteria for Genetic Algorithms," Springer, Berlin, Heidelberg, 2004, pp. 405–413.

[109] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, Jul. 2006.

[110] C. Cortes, C. Cortes, and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, pp. 273--297, 1995.

[111] T. Saethang, D. M. Payne, Y. Avihingsanon, and T. Pisitkun, "A machine learning strategy for predicting localization of post-translational modification sites in protein-protein interacting regions," *BMC Bioinformatics*, vol. 17, no. 1, pp. 1–15, 2016.

[112] Y.-R. Tang, Y.-Z. Chen, C. A. Canchaya, and Z. Zhang, "GANNPhos: a new phosphorylation site predictor based on a genetic algorithm integrated neural network."

[113] A. K. Halder, P. Chatterjee, M. Nasipuri, D. Plewczynski, and S. Basu, "3gClust: Human Protein Cluster Analysis," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, pp. 1–1, 2018.

[114] Y. Xue *et al.*, "GPS-SNO: Computational Prediction of Protein S-Nitrosylation Sites with a Modified GPS Algorithm," *PLoS One*, vol. 5, no. 6, p. e11290, Jun. 2010.

[115] Y. Xie *et al.*, "DeepNitro : Prediction of Protein Nitration and Nitrosylation Sites by Deep Learning," *Genomics. Proteomics Bioinformatics*, vol. 16, no. 4, pp. 294–306, 2018.