Prediction of Protein Subcellular Location using Residue Exposure

A Dissertation

Submitted in Partial Fullfilment of the

Requirements for the Degree of

Doctor rerum naturalium (Dr. rer. nat.)

to the Department of

Department of Biology, Chemistry and Pharmacy

of the Freie Universität Berlin

by

Arvind Singh Mer

Berlin, 2014

1st Reviewer: Prof. Dr. Ernst-Walter Knapp

2nd Reviewer: Dr. Miguel Andrade

Date of defence: 26 March, 2014

“Your Excellency, I am basically a scientist. Clarity of formulation is essential in my profession.”

Spock (in The Mark of Gideon)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The cell is a three-dimensional space separated into different compartments. The functional machinery of the cell - proteins - need to be present at specific cellular compartments so that the cell can function properly. For example, for the correct execution of the citric acid cycle, all necessary proteins are required to be localized in the mitochondria. The same holds for DNA replication, signal transduction and, more generally, for all cellular functions that require protein-protein interactions of one sort or another, since the interacting proteins must be localized in the same cellular location [1, 2]. Knowing the subcellular location of a protein is helpful for designing experimental strategies for its detailed characterization. For example a proteolytic assay might be a good choice for the study of an extracellular protein, while for nuclear localized proteins DNA-footprinting might be a better choice. Correct protein subcellular location plays a vital role in different cellular mechanisms. A relevant example is the asymmetric cell division of stem cells. In *Drosophila melanogaster*, asymmetric cell division of neuroblasts produces a ganglion mother cell (GMC) and another neuroblast. The GMC produces a pair of neurons while the neuroblast goes again through the asymmetric cell division to produce another GMC and neuroblast. The proteins Numb and Prospero, which are crucial for generating asymmetry in neuroblasts, are synthesized and equally distributed throughout the neuroblast cytoplasm. When the neuroblast starts to undergo mitosis, they localize at the basal cortex and initiate the process of asymmetric cell division. In GMCs, Prospero re-localizes to the nucleus and acts as a transcription factor. This localization of Prospero changes the fate of the GMC, which results in the formation of neurons [3, 4]. Similarly, cellular dynamics of many proteins (e.g. NFkB [5, 6]) are subject to proper subcellular localization. Thus, in order to understand cellular mechanisms, it is crucial to have information about the subcellular location of proteins.

The role of protein subcellular localization in diseases is well known [7, 8]. The hereditary kidney-stone disease is caused by the mislocalization of the enzyme alanine-glyoxylate aminotransferase (AGT) to mitochondria instead of peroxisom [9]. Aberrant localization of proteins can lead to deleterious gain-of-function or dominant-negative effects. Hereditary disorders such as nephrogenic diabetes insipidus [10, 11] and retinitis pigmentosa [12, 13] are caused by aberrant localization of G-protein-coupled receptors (GPCRs). In neurodegenerative diseases such as Alzheimer's, Parkinson's, Huntington's and amyotrophic lateral sclerosis (ALS) protein subcellular localization plays a vital role [14–17]. Aberrant cytoplasmic localization of transcription factors and their regulatory kinases such as p53, NF-kB, activating transcription factor 2 (ATF2), cAMP response element-binding (CREB), E2F transcription factor and NF-E2-related factor 2 (NRF2) contribute to degenerating neurons [17, 18]. Several studies have shown that cancer is a disease of cellular pathway deregulation associated with the localization of essential proteins [19–25]. For example, the tumor suppressor proteins p21 and p27 have been found to have oncogenic roles when localized in the cytoplasm instead of the nucleus [26–28].

The knowledge about protein subcellular localization can also be used for therapeutic purposes. The Phosphoinositide 3-kinase/Akt signaling pathway plays a role in cell growth and proliferation and is linked to cancer if deregulated. The pathway can either be inhibited, but this may lead to serious side effects, or another strategy is used which makes use of a compound called perifosine. Perifosine binds to Akt and can decrease its plasma membrane localization. This leads to a partial silencing of the Akt pathway because Akt needs to be present in the plasma membrane to execute its function. This strategy has been tested in clinical trials as a prospective cancer treatment.

While attaining proper protein subcellular localization is a more complex issue in eukaryotic than in prokaryotic cells given that the latter have no membrane bound organelles, proper protein subcellular localization is also important for prokaryotic cells [29]. Knowledge of protein subcellular localization in prokaryotic cells is important for genome annotation and proteome characterization [30, 31]. Furthermore it is also vital for diagnosis and drug development against pathogenic bacteria. For example, the extracellular or secreted proteins can be used as diagnostic biomarkers. Similarly, the surface proteins can act as an important drug target or vaccine component [32, 33].

Given the necessity, several methods for the detection and prediction of protein subcellular location have been developed. Although experimental approaches such as immunolocalization and fluorescent reporter based detection provide reliable determination of protein subcellular location, they are time consuming, laborious and expensive. Furthermore, it is difficult to scale-up and apply these methods in a wide range of organisms and tissues. Thus, such experimental approaches cannot cope with the rapidly growing sequence data. Currently, most protein sequences in databases are the result of translation of hypothetical transcripts derived from genomic sequencing data. Therefore, computational prediction of protein features from their sequence is often used for designing strategies for protein experimental characterization and is also important for genome annotation, interpretation of screens and drug target identification. In particular, the computational prediction of subcellular location from protein sequence information has been attempted mainly using three approaches: search for signal peptides, sequence homology based methods and using amino acid composition of the protein as a proxy for location. Although the performance of signal peptide and homology based methods are reasonable, they cannot be applied to any set of compartments and proteins. For example, our knowledge of signal peptides is incomplete and thus absence of known motifs cannot be used to imply that a protein remains in the cytosol. The homology based methods are limited by the amount of proteins with experimentally verified location. Furthermore, there are many known exceptions to the assumption that similar proteins end up at similar subcellular locations (e.g. the proteins of the Lsg1 family of GTPases [34]). The amino acid composition approaches are based on the hypothesis that the physicochemical properties of the residues of a protein must be somehow coupled to the physicochemical properties of the environment where the protein performs its function; therefore the differences between environments will be imprinted in the protein's amino acid composition. This approach has the advantage that it can be applied to any set of compartments and proteins, provided one has enough data.

Despite several advances, most composition-based methods for the prediction of location are a simple application of machine-learning methods without much biological background and providing little biological insight. Therefore, in this work we first present a detailed analysis of the relation between protein amino acid exposure, residue type and subcellular location. This allows us to establish the fact that physicochemical properties of the residues of proteins are correlated to their subcellular location and that this correlation changes with the exposure of the residue. The focus here is on eukaryotic proteins and three locations: nuclear, cytoplasmic and extracellular. In addition, we will consider the necessity of introducing a fourth class and demonstrate that membership to this class can be predicted: proteins of nucleocytoplasmic localization. This class is not generally taken into account by methods of prediction of location, despite the fact that a large number of proteins are known to shuttle between nucleus and cytoplasm and perform functions in both compartments. We also introduce a two step novel classification approach that uses a support vector machine (SVM) and an artificial neural network (ANN). We will illustrate the usefulness of our novel approach through application of the method to pairs of homologous proteins with different experimentally known location (e.g. two homologous proteins where one is localized to the nucleus and the other to the cytoplasm). The analysis indicates that the method can find the appropriate location in cases where methods using homology would make a wrong inference.

1.2 Overview

Chapter 2 starts with an overview of experimental techniques used for subcellular location detection. Advantages and disadvantages of the most commonly applied experimental techniques are discussed. Computational approaches for subcellular location prediction are introduced in chapter 3. It provides an overview of the main approaches used in computational prediction of subcellular location. Chapter 4 discusses the two main machine learning methods used in this work: SVMs and ANNs.

In chapter 5 we present the analysis of the relation between protein amino acid exposure and subcellular location. The procedure of selecting data for protein subcellular location and integration with protein three-dimensional structure is introduced. The basics of relative accessibility and residue exposure frequency are explained. The chapter closes with a discussion on the residue exposure pattern present in proteins belonging to different subcellular locations.

Chapter 6 addresses the role of different residue exposure ranges in subcellular location prediction. We will also discuss the problems associated with applying machine learning methods to unbalanced data.

After developing the algorithm to predict protein subcellular location using residue exposure, in Chapter 7 we illustrate the significance of our method through its application for the analysis of homologous proteins pairs. The procedure for selecting homologous protein pairs with distinct subcellular localizations is explained. The chapter also describes the comparison of our method with other state-of-the-art subcellular location prediction tools. The results and performance evaluation are discussed in detail.

Chapter 2

Experimental Approaches for Detection of Protein Location

Subcellular localization is a fundamental feature of a protein and correlated with its function. For example a protein localizing in the nucleus can be inferred to be involved directly or indirectly in transcription and gene-expression control. Similarly, proteins localizing in mitochondria may be related to cellular respiration. In prokaryotic organisms, knowing that a protein localizes on the surface of the bacteria can make it a potential drug target [29, 35] or a useful vaccine component [33, 36]. Thus subcellular location analysis has a vital importance in protein characterization. Consequently, for analyzing protein's subcellular location several experimental approaches have been developed. The following section describes a broad overview on experimental techniques of protein subcellular location analysis.

2.1 Immunolocalization

Prediction of protein location via immunological detection methods is one of the most robust methods. This is also one of the first methods of protein location prediction. To analyze protein location, specific antibodies against the native proteins are generated. These antibodies can be used for localization detection using a fluorescent secondary antibody and fluorescence microscopy. A large scale project called the Human Protein Atlas, aims to generating antibodies for at least one protein from each human protein-coding gene [37–39]. Subsequently, using these antibodies the protein distribution at the subcellular level is investigated in different cell lines and tissues [40].

Though immunolocalization methods are robust, they are hard to implement. One of the big challenges in immunolocalization methods is to generate specific antibodies against the protein of interest, which is a very effortful and time-consuming process. Antibody generation also requires sufficient previous knowledge about the protein of interest (e.g. the need to purify the protein) [41]. Due to all these factors immunolocalization methods lack scalability and they are not suitable for genome-wide detection of protein localization. To overcome the laborious process of generating antibodies against native proteins, epitope tagging methods are applied. In this method, rather than generating antibodies against the native protein, a well-characterized epitope sequence is fused to the gene of interest.

The antibody against the marker epitope can then be used for protein location detection. As this method does not require the generation of antibodies against the native protein, a single antibody can be used for protein location detection of several proteins. It provides a useful way for large scale protein location study. There are several variations of the method of epitope-tagging of proteins. In *S. cerevisiae*, epitope-tagging has been successfully applied at whole-genome level [42] and several shuttle vectors are available for epitope-tagging [43]. Gene trap screening and sequencing strategies [44] are able to detect the location of more than 100 proteins, which are mostly localized within compartments of the nucleus, nuclear periphery or in other nuclear foci. Kumar et al. [45] used directed topoisomerase I-mediated cloning and genome-wide transposon mutagenesis strategies to tag the *S. cerevisiae* proteome. The authors claim to have epitope-tagged 60% of the *S. cerevisiae* proteome and subsequently using high-throughput immunolocalization they elucidated the subcellular location of 2744 proteins in *S. cerevisiae*. By combining these results with published location information they further identified subcellular location for about 55% of the *S. cerevisiae* proteome (considering the size of yeast proteome of about 6100 proteins). Interestingly, to detect the subcellular location of *S. cerevisiae* proteome, machine learning methods were applied on the remaining (45%) of proteins. The overall accuracy of this large scale subcellular location study is about 85%. This indicated that computational methods of subcellular location prediction can be complementary to large scale experimental methods.

The immunolocalization technique is also used as rapid *in situ* protein subcellular location detection in plant cells [46]. The protocol takes only a short time (2-3 days) and can be applied to various tissues and embryos of different plant species.

In any immunolocalization experiment the choice of the antibody is crucial for the outcome of the experiment. Furthermore, fixation of the tissue/cells and permeabilization (making a membrane or cell wall permeable through the use of surfactants) without disturbing the tissue and cell integrity requires skills and is a time consuming process. If not done properly, accessibility to epitopes will be low or none, which will lead to wrong results. Epitope tagging of the partial open reading frames (ORFs) can interrupt localization signals. Without these localization signals the protein can end up at the wrong subcellular location.

Using random transposon-tagging, Ross-MacDonald et al. [47] developed economical methods for genomic scale analysis of gene expression and protein location. They generated a collection of over 11,000 strains of yeast *S. cerevisiae* mutants within a single genetic background in which each was carrying an inserted transposon. Using indirect immunofluorescence, the authors analysed over 1300 transposon-tagged proteins. However, during immunofluorescence analysis, two-thirds of all the randomly placed transposon-tags did not yield observable staining patterns. To detect the protein at a specific location sometimes overexpression of the protein is necessary. Though the overexpression of the protein facilitates its detection it can lead to a saturation of the cellular transport mechanisms and to abnormal subcellular localization [48].

2.2 Fluorescent reporter based detection of location

Green fluorescent protein (GFP) is widely used to tag proteins *in vivo* in molecular biology. Due to the special sequence of three amino acids: serine-tyrosine-glycine (sometimes, the serine is replaced by threonine), GFP can create its own fluorescent chromophore [49]. It does not require any exogenous substrates or cofactors for being fluorescent. GFP, first isolated from the jelly fish *Aequorea victoria*, has several variants and can be expressed in a wide range of organisms. While most fluorescent molecules are strongly phototoxic and not suitable for live cell imaging, GFP is usually much less harmful when illuminated in living cells. This makes GFP a very popular reporter for monitoring gene expression [50] and protein localization *in vitro* [51, 52].

GFP has been used in many large-scale and genome wide analysis of protein subcellular location. Using the phage Lambda recombination system, Simpson et al. [53] generated N-terminal and C-terminal GFP fusions of 107 human cDNAs. The monkey Vero cells and human HeLa cells were transfected with these constructs to determine the subcellular location of the corresponding proteins. The authors determined clear intracellular localization to known structures or organelles for about 80% of these proteins.

The fission yeast *Schizosaccharomyces pombe* has been used as a model organism in several genome and proteome wide studies as its genome size is quite small compared to other known eukaryotes [54]. Furthermore, it shares many traits with higher eukaryotic cells.

To search for the intracellular localization of proteins, Ding et al. [52] constructed a GFP-fusion genomic DNA library of fission yeast *S. pombe*. To construct the library, random fragments of genomic DNA were fused in all three reading frames to the 5' end of GFP in a fashion that GFP-fusion protein expression would be under the control of the own promoters contained in the genomic DNA fragments. The yeast *S. pombe* was subsequently transformed by the GFP-fusion library and cells were screened for fluorescence by microscopy. They screened 49845 transformants out of which 728 transformants exhibited GFP fluorescence localization. The localization was in distinct and well-defined intracellular structures such as the nucleus, the nuclear membrane, and cytoskeletal structures. Using plasmid isolation, the authors categorized intracellular location of 250 GFP-fusion constructs.

Efforts were made to determine protein subcellular location at wild-type levels of protein expression or keeping perturbation in protein expression at minimum. Huh et al. [48] constructed a collection of yeast strains expressing full-length, chromosomally tagged GFP fusion proteins. The proteins were tagged at the carboxy terminal end with GFP under their endogenous promoters. To do so the coding sequence of GFP was inserted in-frame immediately preceding the stop codon of each ORF. With this strategy the authors were able to circumvent potential problems of transposon-mediated random epitope tagging [47] as localization signals can be interrupted in epitope tagging of ORFs. Furthermore this method did not require overexpression of proteins. This is an advantage over plasmid-based overexpression of epitope-tagged proteins [45] as the overexpression of proteins can saturate the cellular transportation machinery and can lead to abnormal subcellular localization. The authors defined the subcellular locations for about 75% of the yeast proteome in the GFP library [55] and were able to determine subcellular location for about 70% of previously unlocalized yeast proteome. On comparing their results with earlier done large-scale studies [45, 47] and published location data from the Saccharomyces Genome Database (SGD) [56], authors found high (about 80%) agreement. For the nuclear pore complex, the results were compared with a previous study that used mass spectrometry for nuclear pore complex analysis. The comparison revealed that out of 29 nuclear pore complex proteins identified by mass spectrometric analysis [57], 23 proteins were localized to the nuclear periphery. Similarly, 14 proteins out of 16 spindle-pole-body components identified by mass spectrometry [58] were rightly localized to the spindle pole. In the study the proteins were classified in 22 different subcellular location categories and generated location data for about three-quarters of the yeast proteome.

In a large scale project called 'ORFeome', Yoshida et al. [59] cloned the entire set of protein-coding open reading frames (ORFs) of the fission yeast *S. pombe* for genomic and proteomic level analysis. Using a recombination-based cloning approach, the authors created 4910 ORFs for analysis. By tagging each ORF with YFP (yellow fluorescent protein) they determined the subcellular location for 4431 proteins in *S. pombe.* The authors also did a genome-wide comparison of protein subcellular location between two different eukaryotic organisms. They compared the protein subcellular location data from previous studies on budding yeast *S. cerevisiae* [48] with the *S. pombe* localization data. Interestingly the comparison showed that not all pairs of homologous proteins had the same location. The study also revealed a subset of homologous proteins that are known to be localized in the bud neck in *S. cerevisiae* whereas in *S. pombe* they localized in the septum. The fact that septum and bud neck are structurally distinct makes this discovery more interesting. The analysis also revealed that more than 50% of all proteins are localized in multiple compartments, which is a larger amount than the existing literature would suggest. Crm1 (Chromosome region maintenance protein) is an importin family nuclear export receptor that facilitates nuclear export of proteins having the nuclear export signal (NES) sequence [60]. To identify the proteins whose location is mediated by Crm1, they inhibited Crm1 using leptomycin B [61] and analysed the subcellular location. They discovered 285 proteins whose subcellular location was modified by Crm1 inhibition. On closer analysis it was found that the proteins whose localization is disturbed by Crm1 inhibition and got accumulated in the nucleus, belong not only to the expected location category (like cytosol) but also include proteins that are supposed to localize in other cellular location, including the septum. This indicates that, for the proper localization of these proteins it is necessary to have a transit into and out of the nucleus. Crm1 required the proteins to have a leucine-rich nuclear export signal for nuclear export [62]. It is widely assumed that only proteins with a leucine-rich nuclear export signal can use Crm1 as a localization and transport mediator [61]. Contrary to this, the study revealed that almost half of the proteins that use Crm1 as a nuclear exporter do not have a clear leucine-rich nuclear export signal. This indicates the limitations of our knowledge of the signals used by the cell to sort proteins. How these proteins are able to undergo Crm1 dependent nuclear export without a 'consensus' leucine-rich nuclear export signal? One possible explanation is that these proteins are using the so-called "piggy-back" mechanism of transport in which they associate with other proteins that bear functional nuclear export signals [63]. It may also be possible that our understanding of 'consensus nuclear export signal' is not complete and we may need to further investigate and redefine complex nuclear export signals. The study also reveals the fact that protein subcellular localization pathways are not a simple transportation process from the cytosol to other subcellular location and rather involve complicated cellular processes and several cellular organelles.

Elucidating protein subcellular localization using GFP and fluorescence microscopy has its limitations as it is not possible to distinguish some locations using this method. For example, these methods are not able to distinguish membrane versus mitochondrial lumen or the endoplasmic reticulum [48, 64].

Fusing GFP to the C-terminal may cause mislocalization of proteins. It can interrupt the localization signal sequences at the C-terminal [65] or can cause steric hindrance in protein structure [66], which can result in mislocalization. For example, due to modification of the C-terminus with palmitoyl and farnesyl groups, Ras2 gets localized in the plasma membrane [67, 68]. Tagging the Ras2 gene with GFP at its C-terminal interferes with the C-terminus modification process, which leads to the mislocalization of Ras2 to the nucleus and the cytoplasm [48].

Similarly, many other proteins, specially those known to be localized in peroxisome [69], endoplasmic reticulum [70] and cell wall [71], contain C-terminal targeting signals. GFP tagging at C-terminal of such proteins often shows wrong localization results.

2.3 Subcellular proteomics approach

In addition to immunofluorescence and fluorescent tagging of proteins, subcellular proteomics is a powerful method that analyses the whole proteome of a given subcellular compartment at once. The method is based on the biochemical fractionation of cells to isolate the compartment of interest followed by a further separation of the proteins by gel electrophoresis. The bands are excised and subsequently digested to make protein fragments amenable for analysis by mass spectrometry. Dreger et al. [72] used this method to identify novel integral membrane proteins of the inner nuclear membrane. They found 148 different proteins, among them 19 previously unknown or uncharacterized, by combining 16-benzyl dimethyl hexadecyl ammonium chloride (16-BAC) gel electrophoresis and matrix-assisted laser desorption ionization (MALDI) mass spectrometry.

The largest advantage of this technique is the abolishment of laborious and time consuming antibody production and protein tagging. However, the accuracy of this method crucially depends on the quality of the biochemical preparation of the subcellular compartment. As the fractionation of cells is prone to impurities and can distort the outcome of the mass spectrometry analysis, a validation of the results by immunofluorescence or fluorescent reporter based assays is necessary. Another disadvantage over the previously described methods is the limitation in studying changes in protein localization. If, for instance, a protein shuttles between nucleus and cytosol its fragments can be detected in both the nuclear and cytosolic fraction. This does not clarify whether copies of the same protein reside in both compartments at the same time or the protein shuttles between them. Therefore subcellular proteomics will only provide a static picture of the organelle's proteome at a given point in time, in contrast to e.g. fluorescent tagging of proteins that can unravel the protein trafficking.

Chapter 3

Computational Prediction of Protein Location

While the experimental methods of protein subcellular location prediction provide reliable detection, they are not able to cope with growing amount of genomic and proteomic data. Furthermore in many cases it is not possible to apply the experimental methods. To overcome such problems and to take advantage of the growth of biological data, over the years different computational approaches have been implemented for protein location prediction in eukaryotic and prokaryotic cells. On the basis of their underlying principles these methods can broadly be classified as

- Signal peptide methods
- Homology based methods
- Sequence feature based methods

Some of the modern protein subcellular location prediction tools utilize all of these approaches for high end accuracy. The reason behind this is that there is no one principle that fits all proteins. Moreover, lack of reliable data sometimes can invalidate the underlying principle. This is a frequently encountered problem in homology based methods. In the following sections we describe these principles and their implementation strategy.

3.1 Signal peptide methods

The cell is a crowded place, thousands of proteins are getting produced and transported to various cellular locations. After translation, most proteins in the cell are recognized by the protein sorting machinery and transported to the appropriate destinations in the cell or secreted outside the cell. The protein translocation process often requires helping proteins like chaperons and in case of active transport requires an energy (ATP or GTP) gradient [73–75]. However some proteins can diffuse in and out of an organelle such as the nucleus. In case of the nuclear pore complex, it was generally believed that proteins smaller than $60kDa$ molecular weight are allowed to diffuse through [73, 76, 77]. Recent studies have shown that proteins with a molecular weight larger than $60kDa$ can also diffuse into the nucleus [78].

To enter inside a compartment the protein has to cross the compartment membrane. The proteins whose translocation is facilitated by machinery present at the surface of the compartment need to have localization information that must be recognized by the sorting machinery. This localization information is generally present at the primary sequence level in form of short sequence segments called signal peptides. In particular, signal peptides are 3 to 70 amino acids long sequence motifs usually present at the terminus of many newly synthesized proteins. They are also referred as target peptide, sorting signal or leader peptide. In the process of protein sorting the signal peptides play a critical role and act like a zip code for proteins [79]. By interacting with appropriate receptors at the organelle membrane, they guide the protein transport and translocation process [80, 81]. After entering the appropriate cellular compartment, the signal peptides usually get cleaved by signal peptidases [82, 83]. Various studies have analysed features of signal peptides. For example, the mitochondrial targeting peptides are poor in negatively charged residues, rich in Arg, Ala, Leu and Ser, and form amphiphilic α-helices [84, 85]. Similarly, the chloroplast targeting peptides are rich in Ser and Thr residues but they have less negatively charged amino acid residues (Asp and Glu) [84, 85]. Most signal peptides are known to have a region rich in positively charged amino acid residues at one side, a polar residue rich region at the other side and in between is a hydrophobic residue region [85, 86]. Unfortunately, signal peptides are not always a well-defined linear motif, rather in many cases localization information is contained in vague sequence features. The signal peptide based method of protein subcellular location prediction tries to define and analyses the motifs crucial for protein localization. Using knowledge base and machine learning approaches these methods can predict subcellular location with reliable certainty. Early work predicting signal peptides used simple linear discriminant methods. For example one of the first methods used a weight matrix and simple rules like the residues at -3 and -1 positions from the cleavage site must be neutral and small, to discriminate signal peptide sequences from the non-signal peptide sequence [87]. As the amount of the reference signal peptide data grew, such simple rules did not seem valid in several cases. To overcome these problems non-linear discriminant approaches and machine learning methods such as artificial neural networks (ANNs) [88], hidden markov models (HMMs) [89] and support vector machines (SVMs) [90] have been extensively applied. Nielsen et al. developed the prediction tool called "SignalP", which used an ANN [88, 91]. The initial tools for signal peptide prediction (such as SignalP version 1.0) had limited capability of distinguishing between signal peptides and N-terminal transmembrane helices as both are prominently hydrophobic in nature. In comparison to signal peptides, the transmembrane helices typically contain longer hydrophobic regions and do not have cleavage sites. Thus, complete genome level analysis for signal peptides with those early methods may yield many false positive predictions. Different strategies have been applied to account for this issue e.g. SignalP (version 2.0) used HMMs and submodels for signal anchor [89]. Similarly, tools like Philius [92], Spoctopus [93] and MEMSAT-SVM [94] use the transmembrane protein structure and topology models along with signal peptide models. It is important to mention that the main aim of SignalP (and of other signal peptide prediction tools) is to discover the presence and location of signal peptides and the corresponding cleavage sites in protein sequences rather than subcellular location itself. Nonetheless the tools designed for location prediction using target motif discovery use the same principles and can be used in couple for better prediction and analysis of protein location [95]. One of such specialized tools for signal peptide based subcellular location assignment is "TargetP" [95, 96]. The method uses an ANN and is specialized for eukaryotic proteins. It analyses the protein sequence for the presence of N-terminal presequences such as secretory pathway signal peptides and mitochondrial or chloroplast targeting peptides and assigns one of four classes out of mitochondrion, chloroplast, secretory pathway and "other". Furthermore, using SignalP it also predicts the potential cleavage site for the signal peptide. Analysis with redundancy-reduced protein test sets shows 85% accuracy for plant proteins and 90% accuracy for non-plant proteins. To predict location, another tool, SLP-Local [97], divides the sequence into three regions: N-terminal, middle and C-terminal. These portions are used as feature vector along with di-peptide frequency and other features. Using SVM as classification algorithm, SLP-Local predicts location in one of the 4 classes (nuclear, cytoplasmic, extracellular and mitochondrial) and achieved 87% accuracy for eukaryotic proteins in five-fold crossvalidation. In their analysis the authors concluded that using N and C-terminal parts of the sequence as features is helpful for classification [97]. Tools and methods for detecting organelle specific signal peptides have been also developed. Nucleolar localization sequence Detector (NoD) is one of the first tools for predictions of nucleolar localization sequences in diverse eukaryotes and virus protein sequences [98]. It uses an ANN and reached 79% positive predictive value with 71% sensitivity when testing with experimentally validated nucleolar localization sequences. Moses et al. suggested an HMM-based approach for the prediction of novel nuclear localization signals [99]. Application on a yeast dataset showed a low true positive rate and had 37% success rate. Claros and Vincens performed multivariate discriminant analysis to predict mitochondrial targeting sequences [100]. Several databases also have been compiled for signal peptide related information.

The database NLSdb contains experimentally determined as well as predicted nuclear localization signals using in-silico mutagenesis [101]. SPdb [102] is a database containing experimentally determined and computationally predicted signal peptides. It integrates information from Swiss-Prot and EMBL nucleotide sequence databases.

The signal peptide based methods of subcellular location work better in cases where a known signal peptide is present in the query protein sequence. Unfortunately our knowledge about signal peptides is incomplete. Hitherto unknown signal peptides might be present in a query sequence, in which case the signal peptide based methods will not work efficiently. There is no consensus sequence or rules for all signal peptides. Although there have been several efforts to predict possible novel signal peptides, the high rate of false positive prediction still remains a problem. More than such technical problems, the biological fact that not all proteins have localization peptide sequence makes the method unsuitable for proteome scale location prediction. Recently Ivankov et al. have shown that in *E. coli* only half of the previously estimated proteins contain signal peptides and about 90% of proteins do not have signal peptides [103]. Another important and usually ignored aspect of protein subcellular localization is the ‘piggyback ride’ mechanism. There is a growing amount of evidence suggesting that many proteins, although they do not have specific localization signals, use this mechanism for subcellular localization [104–109]. Subcellular localization for such proteins can not reliably detected via signal based methods. It is safe to say that for a large number of proteins, the localization prediction method solely based on signal peptide is not applicable at all.

3.2 Homology based methods

Homology between protein sequences indicates an evolutionary link, homologous proteins usually share similar properties. Thus the homology between protein sequences can be used for the transfer of annotations. The homology based subcellular location prediction methods try to infer the location from the annotated information of similar proteins [110]. In a large scale study Nair and Rost [111] highlighted the association between sequence homology and similarity in subcellular localization. Authors found out that the subcellular location of proteins is more conserved than expected. The conservation of subcellular localization is true for different cellular compartments and it is analogous to the conservation of structure and functional activity. Further analysis by Yu et al. [112] showed that even at as low as 30% sequence identity, the homology based approach performs well. Pierleoni et al. [113] developed a subcellular location database for eukaryotic proteins using this method. The authors also pointed out the fact that about 68% of the human genome can be annotated using both experimental results and the homology search approach. Interestingly, in *Arabidopsis thaliana* this number is down to only 33%. This is because of the fact that a very low number of *A. thaliana* proteins are annotated with experimentally verified location information.

Identifying sequence homologs is the first step in location prediction via homology. Defining thresholds for finding homologous proteins is a challenging task. Are two proteins having 50% sequence identity homologous? Such questions are difficult to answer and are very much context dependent. Pairwise BLAST and PSI-BLAST are commonly used sequence similarity search tools used for homologous protein identification and subsequent location prediction [114–116]. Different measures of sequence similarity can be used for the assignment of subcellular localization e.g. the BLAST expectation values (EVAL), pairwise sequence identity or HSSP-values [117]. Using BLAST E-value as sequence similarity measure, Horton et al. [118] achieved 83% cross-validated accuracy for 2113 fungi proteins. Authors used 12,771 animal and 2333 plant proteins for further analysis and concluded that the BLAST E-value is sufficient to achieve high (about 94% for animal and 86% for plant) accuracy [118, 119]. Although the results appear trivial if we consider the fact that the datasets used in the analysis were highly redundant and included many well-conserved orthologous protein sequences from SWISS-PROT. Nevertheless the fact that sequence homology can be used for reducing the error rate in localization assignment is well established [120–123]. It has been shown that for subcellular location assignment, the HSSP-value based sequence similarity measure performs better [124]. Unfortunately high sequence similarity itself does not ensure same subcellular location. It has been shown that the shift from the regions of conserved to non-conserved location is very sharp [111]. Furthermore, even a change in only few residues can affect the localization of a protein. In humans, the beta oxidation enzymes are targeted to mitochondria while in yeast the homologous proteins are present in peroxisomes [125]. Another well known example is the Lsg1 family proteins [34]. Members of this family of proteins are present in multiple cellular compartments. The performance of homology-based methods is also questionable in cases where the target protein may have isoforms localized at different places in the cell. In their study Nakao et al. [126] concluded that there are many genes whose protein isoforms have different subcellular location and sequence similarity should be used carefully when predicting protein location.

A major obstacle in sequence similarity based location detection is the requirement of precisely annotated homologous proteins. Given the vast amount of sequences in databases, it is not very difficult to find homologous proteins to a query protein. But the homologous proteins may not necessarily have annotated localization information. Usually localization prediction is required for proteins that already lack well-characterized highly similar homologs. Thus, homology based methods alone are not sufficient for localization prediction of novel proteins.

3.3 Sequence feature based methods

Some properties of proteins are correlated with their amino acid composition [127, 128]. Accordingly, through analyzing the residue composition of a protein it is possible to infer some of its properties. This is the case for the subcellular localization of proteins [129]. Using correlation analysis for nuclear, extracellular, intracellular, integral membrane and anchored membrane proteins, Cedano et al. [130] analyzed the relation between the amino acid composition and subcellular location. In a landmark study Andrade et al. [131] hypothesized that proteins from different locations have characteristic differences in their surface residues. This is because of the fact that different subcellular locations have characteristic physio-chemical environments and during evolution the proteins localizing in a compartment have to adapt to the environment of that particular compartment. As the surface of the protein is in direct contact with the environment, the adaptation will be most imprinted on the surface of the protein. For example, compared to intracellular proteins, the extracellular protein's surface is rich in polar residues. Similarly, extracellular proteins have a relatively low percentage of ionic residues. Proteins localizing in the nucleus are known to have relatively more positively charged residues as the presence of DNA makes the environment of the nucleus highly anionic.

By organizing experimental and computational observations as a collection of if-then rules Nakai et al. [132] constructed a knowledge base of sequence-function relationships. Using this knowledge base they developed an expert system that required only amino acid sequence information for subcellular location prediction. For animal and plant cells authors considered 14 and 17 locations, respectively and used 401 eukaryotic proteins with known location for training and testing. For testing data they achieved 59% accuracy. This system is named as PSORT and later extended as PSORT II [133, 134]. PSORT II uses sequence driven features and k-nearest neighbors (kNN) classifier for location prediction. Horton et al. [134] also compared the kNN classifier method with three other classifiers: a structured probabilistic model, the binary decision tree classifier and the naive bayes classifier. The cross validation results showed that the kNN classifier outperforms other methods. An important conclusion of this study was that for subcellular location prediction problems, domain specific features are much more efficient than sequence homology alone.

Subsequently, several methods were developed utilizing the idea that amino acid composition of proteins is related to their subcellular location. Reinhardt et al. [135] trained neural networks on amino acid composition of proteins. For prokaryotic organisms, 81% overall classification accuracy is achieved while classifying proteins into three subcellular locations. In case of eukaryotic organisms four subcellular locations were considered and a lower accuracy of 66% is reported. The predictor ‘SubLoc’ uses an SVM for subcellular location prediction of proteins from their amino acid composition [136]. The amino acid composition of the proteins were encoded in input vectors of 20 dimensions, each dimension representing an amino acid. For prokaryotic sequences a 3-class classifier and for eukaryotic sequences a 4-class classifier was trained. The cross validation accuracy of 91.4% for prokaryotic organisms and 79.4% for eukaryotic organisms is reported.

Protein sequence based features such as residue pair frequency are easy to calculate and can be a rich source of information. They also provide a large feature space. For example, the di-peptide frequency can result in $(20 \times 20 = 400)$ features and tri-peptide frequency can lead to $(20 \times 20 \times 20 = 8000)$ features. Such features are used as input in different machine learning algorithms for location prediction. Using a Markov model and residue pair probability, 78.7% classification accuracy was achieved considering three location categories for eukaryotic proteins [137]. The three location categories considered were nuclear, extracellular and a mixture of cytoplasmic and mitochondrial. For four separate categories (considering cytoplasmic and mitochondrial separately) the accuracy was 73%. Fujiwara et al. [86] used the amino acid composition and sequence order for subcellular location prediction. The amino acid composition is used to express the global features of the protein. The local features of the protein are represented via the amino acid sequence order. HMMs and ANNs are used for this classification purpose. Zhang et al. [138] used hydrophobic patterns along with pseudo amino acid composition to predict subcellular location and in the jackknife test an accuracy of 73% was achieved. There are several tools which try to predict subcellular location by representing the sequence features in different ways. For example CELLO [139] computes gapped and ungapped amino acid pair composition. A large number of tools combine several based sequence features with other sequence and textual information. One such tool, ESLpred [114, 140] combines n-peptide composition and their physico-chemical properties with PSI-BLAST analysis. Similarly, pTARGET [141] computes amino acid composition based properties and combines them with the pattern of occurrence of Pfam domains. The tool SherLoc [142, 143] is a hybrid method combining 3 different approaches: MultiLoc2, EpiLoc and DiaLoc and exploits amino acid composition, along with motifs information as well as text descriptions from the literature and the SwissProt database. The tool Yloc [144] derives more than 30000 sequence based features such as amino acid composition, pseudo amino acid composition, hydrophobicity, etc. Furthermore, the feature space also includes PROSITE motifs and GO terms from close homologous proteins and other homology based features.

The main advantage of sequence feature based methods is that they can be applied to any set of compartments and proteins. In the protein databases, there is a large number of proteins without known signals, known predicted domains associated to protein locations, or without homology to proteins of experimentally verified protein location. For such proteins the sequence feature based methods are the only reasonable choice.

Chapter 4

Machine Learning

Not long ago ‘Biological discovery’ was considered to take place on the lab bench. But in the last years, high-throughput methods of analysis have changed the face of biology and the era of ‘omics’ has arrived. Whether it is a question of genome evolution or how cancer drugs will affect a particular patient, scientists have to grapple with big data and the amount of biological data is growing exponentially.

The European Bioinformatics Institute in Hinxton, UK, has currently 20 petabytes of data related to genes, proteins and molecules and this amount is increasing every year [145]. In parallel, the questions biologists are trying to address are of increasing complexity. Thus, generating large amount of data is a first step in biological discovery. To gain an insight the data need to be processed, analysed and integrated, which requires sophisticated algorithms and techniques. Machine learning approaches are ideally suited for such analysis and characterization of complex and large amounts of biological data. Machine learning, a branch of artificial intelligence, refers to construction and study of algorithms and systems that can learn from data.

Machine learning methods are data-driven algorithms. In machine learning algorithms, unlike “normal” algorithms it is the data that drives the algorithms to find the best answer. Let’s assume we want to differentiate between apples and oranges. A hypothetical non-machine learning algorithm for this task will try to define what is an apple (or orange). For example it can try to define the geometrical shape of an apple or the amount of red pixels in the photo of an apple, etc. In contrast, a machine learning algorithm would not have such a coded definition of an apple. It will look at various examples of apples and will learn from them what is an apple. Thus, it will learn by examples to be able to distinguish apples from oranges.

Today machine learning is considered as a collection of several different techniques and algorithms though they all share a unifying framework developing since the late 1980s [146]. Biology has a historic relation with machine learning methods. Some of the first machine learning techniques were inspired by biological systems. For example, the perceptron [147] was an attempt to model actual neuronal behavior which later emerged as artificial neural network (ANN) methods. Similarly, the neocognitron [148] and adaptive resonance theory (ART) [149] are inspired by the visual nervous system. Evolutionary algorithms such as genetic algorithms are inspired by natural evolution phenomena such as inheritance, mutation, selection, and crossover [150]. Regardless of their biological inspirations, the use of machine learning methods to solve biological problems started comparatively late. One of the first examples is the use of the perceptron algorithm for the analysis of translation initiation sequences in *Escherichia coli* [151]. In recent years, there has been a large improvement of machine learning techniques and computational power has increased substantially. This led to machine learning becoming a reliable tool for biological discovery in complex and mounting volumes of biological data.

Broadly, machine learning algorithms can be organized into two categories, according to the way the examples are used to train the method, supervised learning and unsupervised learning. Supervised learning works through the generalisation of a set of rules that can be used for the assignment of previously unseen objects to classes based on features. Here, during the model-fitting process the target values of output in the fitted data are known. In supervised learning the goal is to make object-to-class mapping models using currently available objects (and the respective class), so that we can predict the class for unknown objects as accurately as possible. The object-to-class mapping is called training process and this mapping does not need to be absolutely accurate. Prediction of protein location from the amino acid sequence is a relevant biological example of supervised learning. The amino acid related information of proteins can be represented by a feature vector. Based on the feature vector and the respective target values (subcellular location) we can build a model that can predict class membership of new objects based on the available features. Supervised learning methods not only perform discriminant analysis but can also be used for regression analysis.

In contrast with supervised learning, no predefined class labels are available in unsupervised learning. The goal of unsupervised learning is to discover similarities (and dis-similarities) between objects without any external inputs other than the raw data. This happens through clustering the objects in different classes based on the object properties [152]. In this way, unsupervised learning helps to unveil the natural patterns and grouping of data. Unsupervised learning methods can also be used for density estimation or for data dimension reduction. Clustering algorithms such as k-means, or hierarchical clustering are popular unsupervised learning techniques and often used for analysis of high dimension data such as microarray data [153, 154]. Similarly, principal component analysis (PCA) is useful for data dimension reduction and visualization [155, 156]. Self-organizing map (SOM, also called Kohonen map) [157] is also used to produce a low-dimensional, discretized representation of the input space of data and is used in many bioinformatics problems [158–160].

In summary, supervised learning uses labeled data and tries to associate new data with classes; unsupervised learning uses unlabeled data and attempts to define patterns in the data. Most biological data problems boil down to classification, pattern recognition and prediction. The ability of machine learning techniques to cope with high dimensions and nonlinearities of data makes them perfect for biological applications. In the next section, SVMs and ANNs will be introduced which are further used for analysis of protein subcellular location.

4.1 Support vector machines

In machine learning, support vector machines (SVMs), sometimes also called support vector networks, are supervised learning method. They are useful for classification, pattern recognition and regression analysis [161–164]. The theoretical framework related to SVMs has a long history of development starting from the 1960's [165]. The SVMs close to their current form were introduced by Vapnik et al. in 1992 at COLT conference [166]. In 1995 Vapnik et al. further introduced the soft margin classifier [167] and an extension of SVMs for regression analysis [168]. Since then, several further extensions and improvements (e.g. "Kernel Trick") were developed [169] and SVMs have become more popular [170]. In this thesis I utilized the Support Vector Regression (SVR) for the classification of proteins in different subcellular location. Using SVR I calculated the probability value for a protein to be in a location class. The concept of SVR is derived from Support Vector Classification (SVC), thus it is useful to first understand SVC.

4.1.1 Support vector classification

The basic SVM is a non-probabilistic binary linear classifier. In a binary classification problem the objects are labeled with one of two labels. Here we assume a positive class and a negative class denoted as +1 and −1 respectively. Let $\vec{\mathbf{x}}$ denote a vector with $\boldsymbol{n}$ components which implies that $\vec{\mathbf{x}}$ is a point in $\boldsymbol{n}$ dimensional space. The notation $\vec{\mathbf{x}}_i$ will represent the i^{th} vector in the dataset where $i = 1, 2..., M$ and that have $\mathbf{y}_i$, the corresponding class label. For the binary class problem

$$\begin{gathered}\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \vec{\mathbf{x}}_3, ..., \vec{\mathbf{x}}_M \in \mathbb{R}^n \\ \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_2, ..., \mathbf{y}_M \in \{+1, -1\}\end{gathered} \tag{4.1}$$

Here $\vec{\mathbf{x}}_i$ is called a feature vector and $\mathbf{y}_i$ is a class label. The goal is to build a classifier to separate positive instances from the negative ones. In the example plot (figure 4.1), a line can be used to separate the dataset into two classes.

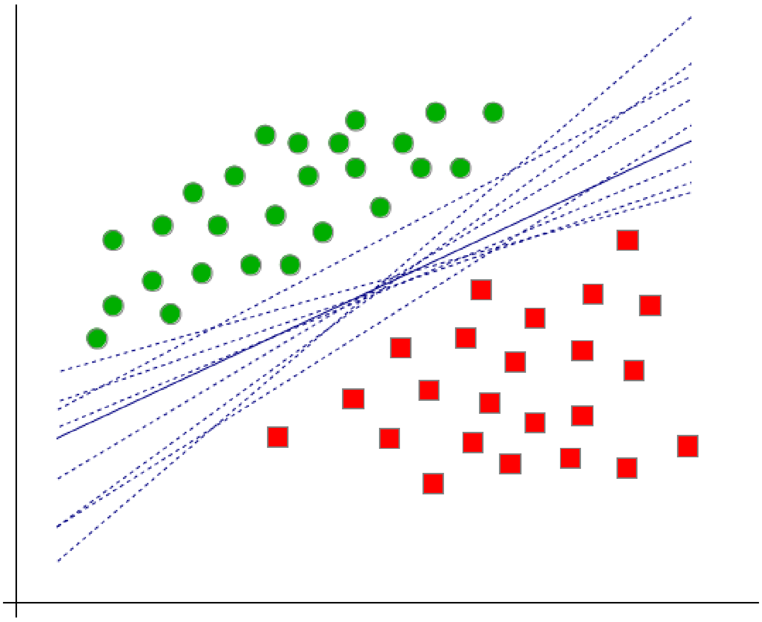


FIGURE 4.1: Classification planes of two separable classes. There can be several possible separating hyperplanes.

This is because the dataset in figure 4.1 is two dimensional. Similarly, if we have a dataset with three dimensions, we require a plane to separate the data; and for $\boldsymbol{n}$ dimensional data we will require an $(\boldsymbol{n} - 1)$ vector subspace, which is called a hyperplane. There can be an infinite number of existing hyperplanes that can separate the data into two classes (figure 4.1). A hyperplane will provide the best classification if the hyperplane not only separates the classes correctly but also does so with largest distance possible to the nearest training data point of any class i.e. with large margin [171, 172]. The data points that are on the boundaries of the class and closest to the optimal separating hyperplane are termed Support Vectors (SV). The optimal hyperplane is the one that maximizes the distance between it and the support vectors. This is called Optimal Separating Hyperplane (OSH). The aim of the SVM is to find such optimal hyperplane that separates the data into two classes.

$$\langle w, x \rangle + b = f(x) \tag{4.2}$$

where w is a vector (known as *weight vector*) in $\mathbb{R}^n$ and b is a scalar called *bias.* To classify all instances, the optimal classifier hyperplane must satisfy the following constraints:

$$\begin{aligned} &\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b \geqslant +1 \text{ if } y_i = +1, \\ &\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b \leqslant -1 \text{ if } y_i = -1 \end{aligned} \tag{4.3}$$

This can be written as a single expression:

$$y_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) \geqslant 1 \tag{4.4}$$

The optimal hyperplane can be computed by solving the following optimization problem:

$$\begin{aligned} &\text{Minimize } \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to: } y_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) - 1 \geqslant 0, \\ &\text{for } i = 1, 2, ..., M \end{aligned} \tag{4.5}$$

where $\|\mathbf{w}\|$ is the *norm* (or the length of $\vec{\mathbf{w}}$). By minimizing $\|\mathbf{w}\|^2$ we maximize the margin. The set of formula above is a convex quadratic programming (QP) optimization problem and is called the primal formulation of linear SVMs. For this problem the optimal solution can be obtained such that it satisfies equation 4.4, while the distance from support vectors is as small as possible. Using the Lagrange multipliers, we can recast this problem in the *dual formulation.*

$$\begin{aligned} &\text{Maximize } \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j \\ &\text{subject to: } \alpha_i \geqslant 0 \text{ and } \sum_{i=1}^{M} \alpha_i y_i = 0 \\ &\text{for } i = 1, 2, ..., M \end{aligned} \tag{4.6}$$

The variable α_i can be interpreted as the contribution of the i^{th} training example to the final solution. The vector $\vec{\mathbf{w}}$ can be defined as

$$\vec{\mathbf{w}} = \sum_{i=1}^{M} \alpha_i y_i \vec{\mathbf{x}}_i \tag{4.7}$$

and now the solution can be written as:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{M} \alpha_i y_i \vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}} + b\right) \tag{4.8}$$

The data points $\vec{\mathbf{x}}_i$ for which $\alpha_i \geqslant 0$ are the support vectors. The large margin hyperplane is defined only by these data points i.e., support vectors and other data-points do not affect the hyperplane. In the dual formulation of the SVM (equation 4.7 and 4.8) we do not require the original data, rather we need to access only the dot product for optimization. Also in the dual formulation, the number of free parameters does not explicitly depend on the number of variables but is bounded by the number of support vectors. Thus, the dual formulation transforms the optimization problem in M variables, where M is the size of the training data. This is especially useful in solving problems with high dimensions as it can save us from the curse of dimensionality [173]. For example for a microarray dataset which contains information on 100 patients and 10000 genes, we need to optimize only up to 100 parameters. Caruana et al. [173] has shown that generally SVMs perform well for high-dimensional data.

The above formulations enforce that all data-points are out of the margin. This is called *hard margin*. This kind of classifier makes sure that all input examples are correctly classified and give zero training error. Consequently, the *hard margin* works only for linearly separable data. Moreover in case of the *hard margin* classifier, the outliers can affect the performance [174]. To overcome this problem, some data-points can be allowed to be misclassified and also a greater margin can be achieved. The result will be a *soft margin* classifier that gives a non-zero training error. The soft margin SVM can achieve better performance than the hard margin and is less likely to overfit. A simple approach of *soft margin* is to assign a *slack* *variable* ξ to each instance. Thus we can rewrite equation 4.4 as

$$\begin{aligned} & y_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) \geqslant 1 - \xi_i \\ & \text{for } i = 1, 2, ..., M \end{aligned} \tag{4.9}$$

where $\xi_i \geqslant 0$ is a slack variable that allows a data-point to be misclassified. From equation 4.5, the new formulation becomes

$$\begin{aligned} & \text{Minimize } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi_i \\ & \text{subject to: } y_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) \geqslant 1 - \xi_i \\ & \text{for } i = 1, 2, ..., M \end{aligned} \tag{4.10}$$

Similarly the dual formulation in equation (4.6) becomes

$$\begin{aligned} & \text{Minimize } \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j \vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j \\ & \text{subject to: } 0 \leqslant \alpha_i \leqslant C \text{ and } \sum_{i=1}^{M}\alpha_i y_i = 0 \\ & \text{for } i = 1, 2, ..., M \end{aligned} \tag{4.11}$$

The variable C works as a control mechanism for the *slack variable* ξ. By having a small value of C, misclassifications are allowed during the training. In case of a very large value of C a large penalty will be assigned to errors and the soft margin SVM will behave similar to a hard margin SVM.

4.1.2 Kernels for nonlinear data

The formulation of the SVM in the previous section is restricted to only linearly separable data. If a classification algorithm is able to handle only linearly separable data, its usefulness would be quite limited as many real world problems are not linearly separable. For example consider the data in figure 4.2. Intuitively, there is some pattern in this data but ordinary formulation for linearly separable data (e.g. equation 4.11) will not able to adequately recognize it.

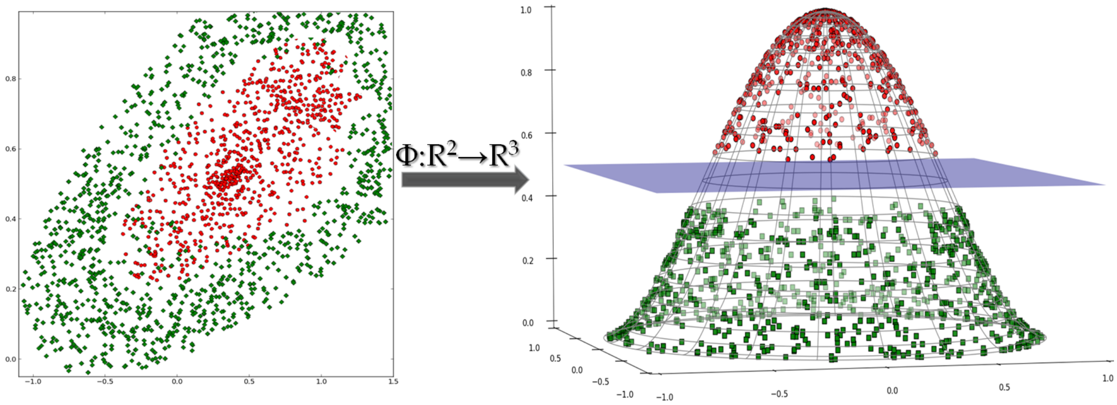


FIGURE 4.2: Classifying data using the kernel trick.

This raises the question whether linear SVM can be extended to the construction of non-linear decision functions for classification of nonlinearly separable data. The straightforward way to deal with this problem is to map the features to another feature space called kernel methods. The basic idea of kernel methods is to first map the data using some non-linear mapping function to a higher-dimensional space and then apply the linear algorithm as before in the higher-dimensional space. Using the non-linear function ϕ for mapping, equation 4.2 can be written as

$$\langle w, \phi(x) \rangle + b = f(x) \tag{4.12}$$

Since the existing data feature space is mapped to a higher-dimensional space, this can substantially increase the number of features to consider, which can be problematic. The kernel methods make sure that the number of features increases only linearly with the size of the data and feature explosion does not occur. They do so by avoiding the explicit mapping of data features to higher dimensional space. Equation 4.7 can be redefined for using the mapping function ϕ as

$$\vec{\mathbf{w}} = \sum_{i=1}^{M} \alpha_i y_i \phi(\vec{\mathbf{x}}_i) \tag{4.13}$$

Similarly equation 4.8 becomes

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{M} \alpha_i y_i \phi(\vec{\mathbf{x}}_i) \cdot \phi(\vec{\mathbf{x}}) + b \right) \tag{4.14}$$

As indicated earlier, the resulting feature space can be very high-dimensional, which should be avoided. We can define the *kernel function* $k(x, x')$ as

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \tag{4.15}$$

Here computation can be done efficiently as we do not need to know explicitly the mapping function ϕ, rather only defining the kernel function $k(x, x') : \mathbb{R}^n \times \mathbb{R}^n \to R$ is sufficient. Furthermore, mapping into very high dimensional space can be avoided. There are certain conditions for a function to be a kernel function. Not every function $R^n \times R^n \to R$ can be a valid kernel. To be a useful kernel a function has to satisfy the Mercer conditions, otherwise the resulting quadratic problem may not be solved [170].

Several different kernel functions have been proposed [175, 176]. Some of the commonly used kernels are:

Linear Kernel

$k(x, y) = x^T y + c$

Polynomial Kernel

$k(x, y) = (\alpha x^T y + c)^d$

Radial Basis Function Kernel

$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$

Sigmoid Kernel

$k(x, y) = \tanh(\alpha x^T y + c)$

In a practical classification problem which kernel to use depends upon the type of data in hand. Kernels have been developed for specific types of data i.e. kernels for sequences [177] or graphs [178, 179]. In case of small data size the nonlinear or complex kernels may lead to over-fitting. A rule of thumb is to start with a simple linear kernel and try other kernel functions if sufficient classification accuracy is not achieved. In case of complex problems, combining different kernels can also be useful.

4.1.3 Probabilistic output from SVM

The standard SVM classification model discussed above is non-probabilistic. From a training set in which each instance is marked as belonging to one of two classes, an SVM training algorithm can build a model that will assign the future examples to one of the two classes. The assignment of a class to an unseen instance will be non-probabilistic, i.e. the classifier will only tell whether the instance belongs to a class or not. It will not produce the posterior probability for the resulting class. The probabilistic output from a classification algorithm is a richer and more expressive formulation of the underlying data pattern. It also enables the post-processing and interpretation of the output.

To map the unthresholded SVM outputs into probability values, several formulations have been proposed, for example feature space decomposing [180], Platt's method [181], SVM binning [182], Isotonic Regression [183] and recently Inductive Venn Predictor [184]. Over the years Platt's method [181] has become the standard for calculating the probabilistic output for SVMs. Libsvm [185], the software library used for SVM based classification in this work has also adopted Platt's method for probabilities output calculation [186]. For predicting the posterior class probability $Pr(y = 1 \mid x)$, Platt's method approximates it by a sigmoid function

$$\begin{aligned} &Pr(y = 1 \mid x) \approx P_{A,B}(f) \equiv \frac{1}{1 + exp(Af + B)} \\ &\text{where } f = f(x) \end{aligned} \tag{4.16}$$

The optimization of parameters A and B is done so that they minimize the negative log-likelihood of the training data. Different optimization methods can be used for this purpose. In the original work Platt uses Levenberg-Marquardt optimization to solve this. Lin et al. [186] proposed a more robust algorithm by using Newton's method with backtracking and implemented it in Libsvm (since version 2.6). Experimental analysis has shown that Platt's algorithm required more iterations and does not produce solutions in some cases, whereas the efficiency and robustness of the Lin et al. [186] method has been demonstrated and has been widely adopted. We utilized this feature of Libsvm for calculation of in class probability of proteins in case of binary classification.

4.2 Artificial neural networks

Artificial Neural Networks (ANNs) are a supervised machine learning method. The concept of ANNs is inspired by the working of the human brain. Through the long course of evolution the human brain has developed remarkable capacities of learning and processing information. It can process complex and non-linear data and can recognize patterns. As a system, the brain is robust and adaptive with generalization ability which helps in the prediction for future cases. In case of numeric computation the modern computers can easily outperform the human brain. However, the human brain is far superior to von Neumann machines in solving deep perceptual problems. The reason of superiority of the human brain is because it differs from a computer at a fundamental architecture level. Compared to von Neumann architecture of computers, computations in the brain are done by a highly connected network of neurons. This raises the question "based on brain computation architecture, can we create intelligent systems?" ANNs are such an attempt to use the organizational principles of the brain for the creation of machine learning systems.

The very first attempt in this direction was made by McCulloch and Pitts in 1943, when they modeled neurons as a switch that remains active or inactive depending on input from other neurons. In the 1960s Rosenblatt, Minsky and other researchers developed neuron models called "perceptrons", that have similar properties like biological neuron networks and can learn and do some pattern recognition. After Minsky and Papert's work in the 1960s [187], showing that simple perceptrons can solve only linearly separable and limited classes of problems, enthusiasm in the field damped. Interest in ANNs began to resurge in the early 1980s. The Hopfield energy approach [188] and the back-propagation algorithm

[189] were some of the major developments. Since then ANNs have been extensively used in solving many complex real-world problems. They have emerged as an attractive choice because of their remarkable information processing, generalization and learning characteristics.

4.2.1 Biological and artificial neural networks

In this section we present a generalized explanation of brain activity which played a fundamental role in the development of ANNs and other neurocomputing approaches. In the nervous system, neurons are the basic building blocks that process information. The human nervous system is composed of billions of neurons of various shapes, sizes and types [190]. A typical biological neuron has three main functional units - cell body or soma, dendrites and axon. The cell body contains nucleus, cytoplasm and the molecular machinery required for the functioning of the cell. From the signaling and information processing viewpoint, the dendrites and axon are the most important part of the neuron. The dendrites receive signals from other neurons. They appear as tentacles sprouting from the soma. The axon is a long projection of the neuron and it passes the signal to other neurons. Thus, in a neuron the dendrites act as receivers and the axon as the transmitter. As illustrated in figure 4.3, the axons eventually branch into collaterals, which are connected to other neurons via synapses. The synapse, a microscopic gap be-

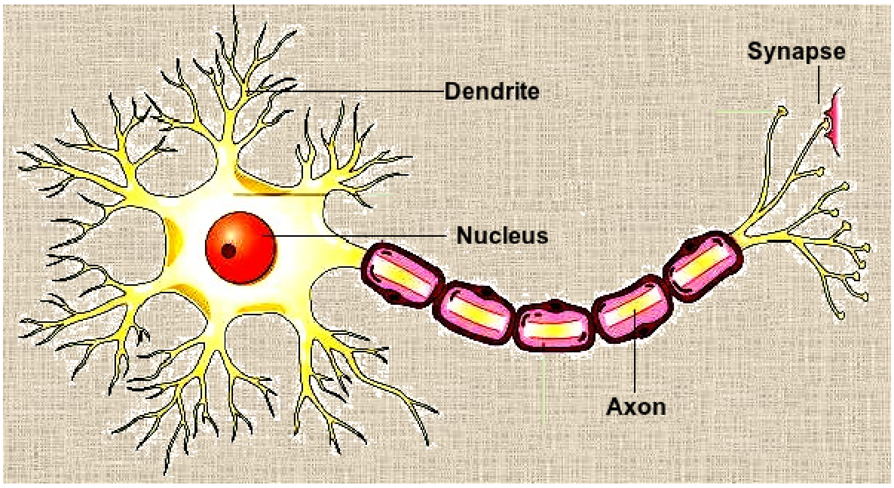


Figure 4.3: Structure of a typical neuron.

tween a neuron's axon and another neuron's dendrite, transmits the signal from one to the other neuron. The impulse travels from the dendrites and the cell body towards the pre-synaptic membrane. The arrival of the impulse at the pre-synaptic end causes vesicles to release neurotransmitters, which diffuse towards the post-synaptic membrane. Once the neurotransmitter binds to receptors at the post-synaptic membrane of a dendrite of another neuron, it causes an action potential in the postsynaptic neuron. The intensity of the signal passed to the other neuron depends on several factors like the intensity of the incoming signal and the threshold of the receiving neuron and, more importantly, the synaptic strength. As the signal passes through the synapse, it can adjust the synaptic strength and thus the synapse can learn from the signal. In essence, the neurons are part of a highly connected network. For example, in the human cerebral cortex, which contains at least 10^{10} neurons, each neuron is connected to 10^3 to 10^4 other neurons via synapses, making in total around 10^{14} synaptic connections [191]. Due to their massively connected nature, neurons can receive and send a large number of signals simultaneously. This also implies that the brain runs parallel processes. At the conceptual level, the neurons in an artificial neural network try to mimic

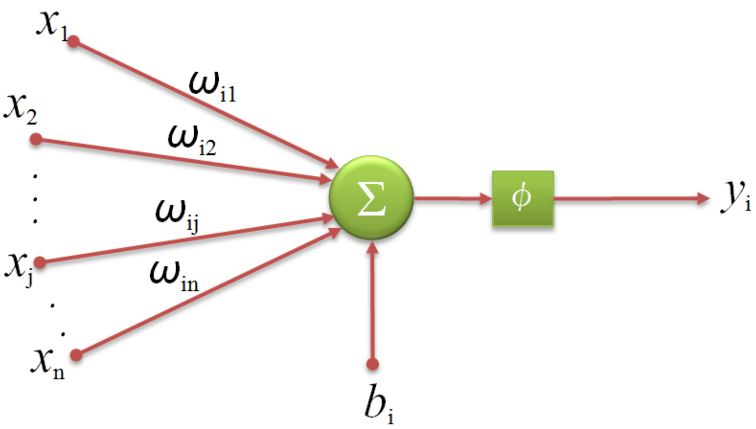


Figure 4.4: Schematic representation of an artificial neuron. The artificial neurons are the basic processing unit of an ANN.

the behavior of actual neurons. The nodes in an artificial neural network represent neurons, while the connections between the nodes can be considered as axons and dendrites. The connection weight is similar to the synapse activity and the threshold acts same as soma in actual neurons. A biological neural network learns by adjusting the synapse activity. Similar to this, the ANN learns by adjusting the weight.

In an ANN an artificial neuron receives input from its environment. It can receive several input signals and integrate all signals in a specific way to produce a combined value. The neuron fires according to this combined value and activation function. For example a neuron with a binary threshold activation function passes the signal to other neurons only when the combined value is greater than a particular threshold. This binary threshold model was first proposed by McCulloch and Pitts [192]. If the McCulloch-Pitts neuron receives n input signals it generates the output of 1 if the sum of input signals is above its threshold. The output y of McCulloch-Pitts neuron can be described as

$$y = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} w_{ij}x_j \geqslant b \\ \\ 0 & \text{if } \sum_{j=1}^{n} w_{ij}x_j < b \end{cases} \tag{4.17}$$

where x is the input signal, w is the synapse weight and b is the threshold. The sign of the synapse weight w determines whether a synapse is excitatory or inhibitory. The positive (+) weight value represents excitatory synapses while negative weights represent inhibitory synapses. The simple artificial neuron described above can learn concepts. The weight values can be adjusted to learn to respond with True or False (1 or 0) for inputs presented to it. Such a system is also called Perceptron.

The McCulloch-Pitts neuron can be modified and generalized in various other forms. For example the activation function can be changed from a binary threshold function to a sigmoid or gaussian function. Table 4.1 shows different types of the activation functions.

In ANNs, the sigmoid function is a frequent choice. As indicated in table 4.1 the sigmoid function has a somewhat identical property to a step function, with the additional region of uncertainty. Compared to other activation functions, the input-output relationship of a sigmoid function is closer to biological neurons [193]. There is a wide variety of sigmoid functions such as the logistic sigmoid and hyperbolic tangent sigmoid functions. The logistic function is the standard sigmoid function and is defined as

$$\phi(x) = \frac{1}{1 + e^{-\beta x}} \tag{4.18}$$

where β is the slope parameter. It is easy to calculate the derivatives of sigmoid functions, which saves time and resources during certain training algorithms.

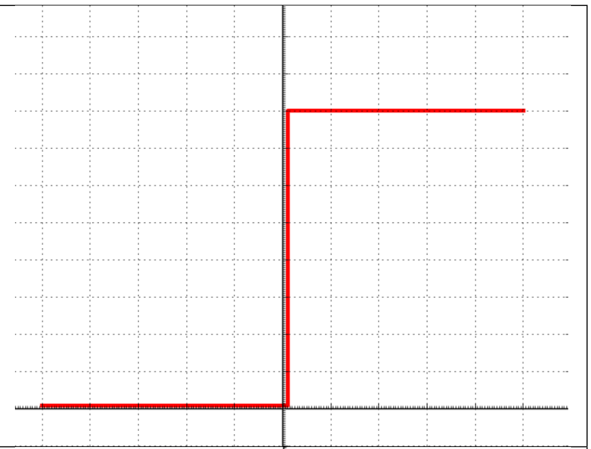
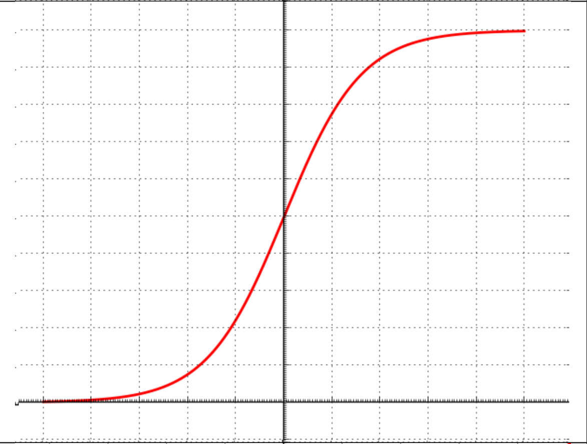
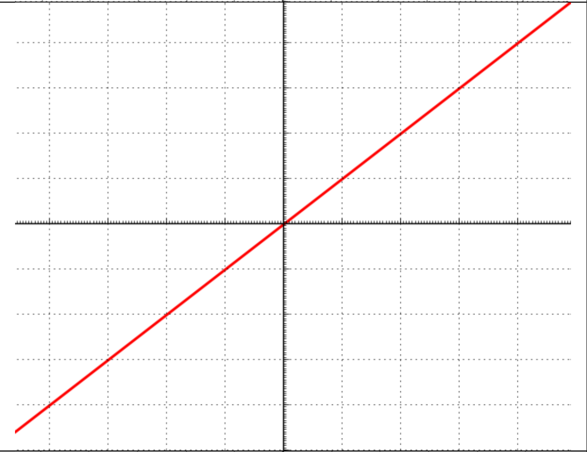
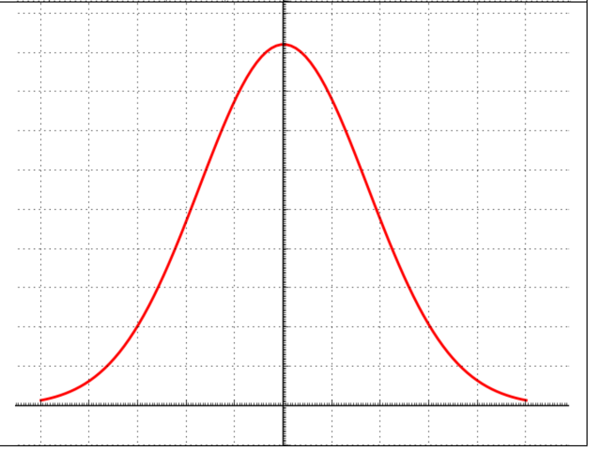
Step Function $\phi(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geqslant 0 \end{cases}$	
Sigmoid Function $\phi(x) = \frac{1}{1+e^{-\beta x}}$	
Identity Function $\phi(x) = x$	
Gaussian Function $\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$	

TABLE 4.1: Examples of activation functions commonly used in ANNs.

4.2.2 Multilayer feed-forward neural networks

The architecture of neural networks can be defined as a weighted directed graph where artificial neurons are the nodes and connections between the neurons are weighted directed edges. Considering this graph theory based architecture, ANNs can be categorized as:

- Feed-forward neural networks
- Recurrent neural networks

In the feed-forward neural networks the nodes (neurons) do not form a loop (directed cycle). Thus the information flow in a feed-forward neural network is only unidirectional. The feed-forward neural networks are static in nature. Accordingly, given an input they generate only one set of output values. Moreover, these output values do not depend upon the network state or the previous input values and they can be regarded as memory-less [194]. On the other hand, the recurrent neural networks have loops or cyclic paths. Because of these feedback loops they are also called feedback neural networks and they are dynamic in nature. In the recurrent neural networks, the output not only depends on the input but also on the state of the network. Thus the network has memory and it can use the internal memory for processing the input. Although recurrent neural networks can be very powerful [195, 196], the theoretical and practical difficulties have currently prevented their practical applications.

For the classification purpose, multi-layered feed-forward neural networks are the most commonly used artificial neural networks. In multi-layer feed-forward neural networks there are no connections between the neurons in the same layer [197]. Moreover, no feed-back connections exist between the layers. The nodes (and the layers) in such neural networks always feed the signal forward. In such networks there is an input layer, an output layer and in between there are hidden layers. The input values are fed in the network via the input layer, which passes it to the hidden layer. After processing, the hidden layer forwards the signals to the next layer. Thus, in a feed-forward neural network composed of three layers, an input layer, a hidden layer and an output layer (figure 4.5), signals are passed from input to hidden and then to the output layer. The hidden layer processes the signal according to connection weights and the activation function (e.g. equation 4.18). Similarly the signal received from the hidden layer is processed by the output layer.

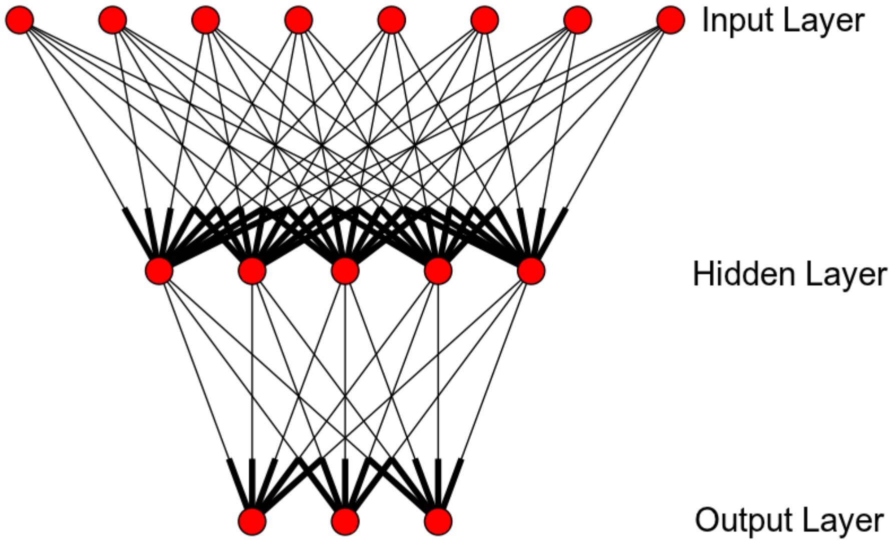


FIGURE 4.5: A typical three layered feed-forward neural network.

The multilayer feed-forward neural networks can solve nonlinear classification problems via making complex decision boundaries.

4.2.3 Learning

ANNs can learn by updating connection weights. In a usual learning process, the training patterns are presented to the network and used to update the connection weights to try to improve performance. ANNs can be trained using different learning paradigms. Unsupervised learning is used for exploring the underlying structure of data, clustering or organizing the patterns. Self-organizing maps (SOMs), also called Kohonen map, are one relevant example. They produce a low-dimensional and discretized representation of the input data, which is useful for visualizing high-dimensional data and multidimensional scaling [157, 198]. Supervised learning approach is commonly used for classification purposes. Here, ANNs are presented both with the training set and with the corresponding output values, based on which the network tries to optimize the connection weights to generate answers as close as possible to the desired output values.

For training of ANNs, the choice of the learning algorithm depends on factors like sample complexity and computational complexity [194]. The error back-propagation learning algorithm is commonly used as a training algorithm. The back-propagation algorithm works on the principle of error-correction. Upon presenting input data, the network generates an output d. This output of the network d may be different from the desired value. The error-correction principle uses the difference between real and predicted values $(y - d)$ to modify the network connection weights to minimize the difference.

Let us consider a feed-forward neural network which has $\boldsymbol{\theta}$ layers after the input layer. In this network a layer $\boldsymbol{l} \in 1, 2, ..., \theta$ has $\boldsymbol{n}$ nodes. The layers before and after $\boldsymbol{l}$ can be represented as $\boldsymbol{l - 1}$ and $\boldsymbol{l + 1}$ respectively. Thus the total number of possible connections to the layer $\boldsymbol{l}$ will be $(n_l \times n_{l-1})$ and there will be $(n_l \times n_{l+1})$ connections from the layer $\boldsymbol{l}$ to the next layer. Each connection has the weight $\boldsymbol{\omega}_{ij}$ where i is the node in layer l and j is the node in layer $l - 1$. Thus node j can be considered as the source node residing in layer $l - 1$ and node i as the destination node residing in layer l . The neuron i on layer l calculates the integrative value $\boldsymbol{\zeta}$ of all incoming signals as

$$\zeta_i^l = \sum_{j=1}^{n_{l-1}} \omega_{ij} x_j^{l-1} \tag{4.19}$$

where x^{l-1} is the incoming signal from the layer $l - 1$ and ω_{ij} is the weight of the connection between neuron i and j . The output value of this neuron will be calculated using the corresponding activation function ϕ (see table 4.1). The sigmoid function (in equation 4.18) is one of such common activation functions. Thus the output value of neuron i, represented as y_i will be

$$y_i^l = \phi(\zeta_i^l) \tag{4.20}$$

Using this equation the output value for each neuron is calculated in a layer and the signal is propagated to the next layer. At the output layer the output signal of the network $\boldsymbol{y}$ is compared with the desired value $\boldsymbol{d}$ and the error signal $\boldsymbol{\xi}$ is calculated

$$\xi = d - y \tag{4.21}$$

By propagating the output layer's error signal ξ back to the network, the back-propagation algorithm tries to optimize the network connection weights. However it is not possible to calculate the error signal for a layer other than the output layer. Thus, the error signals are propagated in a specific way such that the output signal of the neuron is the input signal and the current connection weight is used for error propagation. For the neuron i at layer l the back-propagated error signal will be

$$\delta_i^l = \phi^{'}(x_i) \sum_{k=1}^{n_{l+1}} \omega_{ik} \delta_k^{l+1} \tag{4.22}$$

where ω_{ik} is the connection weight between neuron i at layer l and neuron k at layer $l+1$ and $\phi^{'}$ represents the derivative of the activation function 4.1. This formula allows us to calculate the error signal for each neuron in the network. In the next step the algorithm modifies the connection weights in proportion to the calculated error signal. The change in connection weights $\Delta\omega$ is

$$\Delta\omega_{ik} = \eta \delta_i^l y_i \tag{4.23}$$

Where η is the learning rate. The algorithm uses $\Delta\omega_{ik}$ to adjust the weight ω_{ik} as

$$\omega_{ik} = \omega_{ik} + \Delta\omega_{ik} \tag{4.24}$$

and modifies all the connection weights in the network back up to the input layer. This is the back-propagation step. In the next step, a new data point is presented to the network and the algorithm repeats the steps again. The steps are repeated until the error at the output layer drops below a specified threshold. Another criteria to stop training is when no significant change in error value is observed (error value converges) or a maximum number of iterations is reached.

The back-propagation algorithm uses the gradient descent method for finding the minimum of the error function in weight space. Consequently, the back-propagation algorithm may converge to a local minimum. If there is only one minimum, the hill climbing based gradient descent can cope with the local minima problem. In practical problems, the error surface is usually rough and can contain several local minima and maxima. Depending upon the error surface, gradient descent can be sensitive to the starting point and may lead to local minima. In summary the convergence to a global mimima is not guaranteed when using the back-propagation algorithm. To overcome such problems, several variants of back-propagation algorithms have been proposed, for example using genetic algorithm [199] or swarm optimization techniques [200] together with the back-propagation algorithm.

The back-propagation algorithm is an iterative process. The learning and convergence of error can be very slow and may require a great deal of resources and time. Time and resources required by back-propagation algorithms also depend on the network architecture. For example a fully connected network will have more connections (and parameters) to optimize compared to a partially connected network. During implementation, by using programing techniques like multithreading, cache optimization, etc. convergence times can be reduced.

Chapter 5

Residue Exposure and Subcellular Location

5.1 Motivation

The functional properties of proteins are associated with their subcellular location. Therefore, predictions of protein location can facilitate the study of protein function and characterization. As described in chapter 3, the computational prediction of subcellular location from protein sequence has been attempted using mainly three approaches. The signal peptide based approach tries to identify sorting signals present in the sequence. Although signal peptide based methods show reasonable accuracy in many cases, their performance differs widely between compartments. This is because of the fact that our knowledge about the signals leading proteins into different compartments is limited. The sequence homology based approach considers proteins with sequence similarity to be localized at the same subcellular location. However, the assumption that homologous proteins have similar subcellular locations is not always correct. Exceptions for this rule (e.g. the proteins of the Lsg1 family of GTPases [34]) are growing in number as we gather more knowledge about biological systems. Moreover, similar to signal peptide based methods, the homology based approach is not applicable to all proteins. A third way for the prediction of protein subcellular location uses the general observation that amino acid composition of proteins and protein subcellular location are related. A major advantage of this approach is that it can be applied to any protein and any compartment. Over the years, several tools have been developed utilizing the amino acid composition based approach for protein subcellular location prediction. However, despite several advances these methods have a mediocre performance for several reasons that we will discuss later.

In a landmark study, Andrade et al. [131] have shown that amino acid exposure influences the amino acid composition of proteins in different compartments and inferred that using this property could improve location prediction. The rationale was that differently exposed residues have different evolutionary pressures to mutate towards specific amino acid types whose side chains have physicochemical properties that agree to the subcellular location where the protein performs its major activity. Since the publication of this previous work, much data on protein structures and experimentally verified protein locations has been deposited in public databases.

With the objective of predicting subcellular location from sequence information only, we present a novel analysis of the relation between protein amino acid exposure, residue type and subcellular location, which takes advantage of recent experimental data. Our focus is on eukaryotic proteins and three locations: nuclear, cytoplasmic and extracellular. In addition, we will consider the necessity of introducing a fourth class and demonstrate that this can be predicted: proteins of nucleocytoplasmic localization. This class is not generally taken into account by methods of prediction of location, despite the fact that a large number of proteins are known to shuttle between nucleus and cytoplasm and perform functions in both compartments [201, 202].

Through careful filtering and data integration, we created a reliable high quality dataset and explored the relationship between residue exposure and protein subcellular location. To gain insight into the relationship between amino acid exposure and environment, we performed frequency distribution and PCA analysis on the data set. Thus, using detailed and systematic analysis we showed that proteins belonging to different subcellular locations have distinct residue expose patterns.

5.2 Selection of proteins with known structure and location

The Universal Protein Resource (UniProt) is a comprehensive and freely accessible resource for protein sequence and annotation data [203]. It is known for its high-quality information, minimal level of redundancy and high level of integration with other databases. The subcellular location information of proteins is a fundamental part of UniProt annotation since its creation. The growing quantity and complexity of location information led to some major improvements in the UniProt subcellular location controlled vocabulary, around 2007 (UniProt release version 12.4) and in 2008 (UniProt release version 12.7). The current release version of UniProtKB uses controlled vocabulary to describe the subcellular locations and membrane topologies of proteins under the line “SUBCELLULAR LOCATION” and provides other relevant information.

As our focus is on eukaryotic proteins, we first obtained all eukaryotic protein-identifiers (IDs) from UniProtKB/Swiss-Prot database (release-2012_05). Furthermore, to improve data quality, all unreviewed records from UniProtKB were removed. The resulting protein-IDs were mapped to the corresponding location information defined as ontology terms for subcellular location in the UniProt records. For several proteins for which there is no experimentally verified location information, UniProt provides location information derived from other resources. In the subcellular location annotation UniProt record field this information is described as “by similarity”, “probable” or “potential”. Thus, to select only the proteins with experimentally verified location annotation, all the proteins containing such terms in the subcellular location annotation were removed from the dataset. The proteins that are known to have undergone the post-translational modification process of glycosylation are discarded as the glycosylation can affect the protein’s surface charge and other properties [204–206]. In the next step, all the proteins containing location terms other than nuclear, cytoplasmic or extracellular were removed from the dataset. Thus, a protein having an annotation such as "mitochondria, nucleus" is also discarded as the focus is only on proteins within the above three categories. Interestingly, there is a significant number of proteins annotated as nuclear as well as cytoplasmic (figure 5.1). These are proteins that can shuttle between nucleus and cytoplasm.

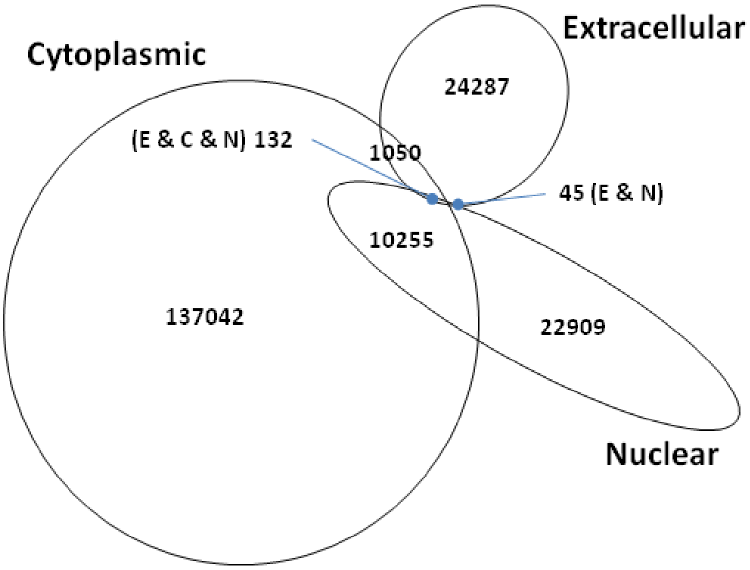


FIGURE 5.1: Venn diagram of eukaryotic proteins exclusively found in three location categories. A significant number of proteins are found both in the cytoplasm and in the nucleus.

This led us to include another location class "Nucleocytoplasmic" in our analysis for the proteins which can have such a dual location. The selected eukaryotic proteins were mapped to the available protein structure entries in the Protein Data Bank (PDB) database. For many proteins, the mapping to the PDB database is not one to one. If multiple PDB entries were available for a sequence we selected the PDB id corresponding to the longest sequence fragment.

Small protein sequences might not have enough residues for doing statistics on their exposed residues and they can increase the noise during the analysis. For this reason, we discarded sequences shorter than 150 amino acids. We ended up with a

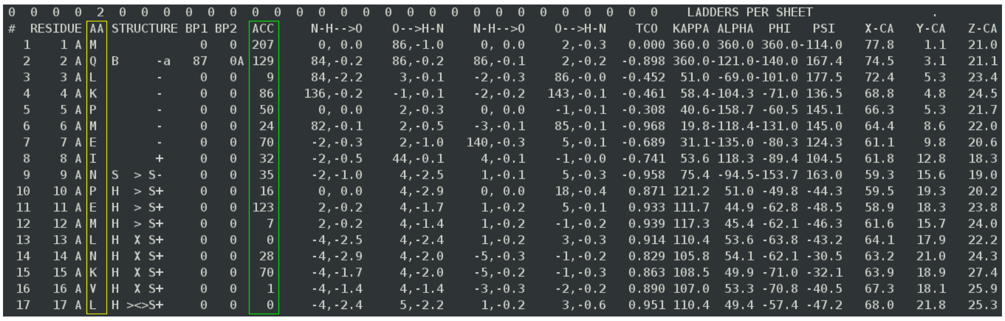
Location	Proteins
Nuclear	336
Nucleocytoplasmic	347
Cytoplasmic	543
Extracellular	132
Total	1358

Table 5.1: Number of proteins with PDB information

total of 336, 347, 543 and 132 proteins for nuclear, cytoplasmic, nucleocytoplasmic and extracellular locations, respectively, for a total of 1,358 proteins (Table 5.1).

5.3 Computation of relative accessibility and residue exposure frequency distributions

For assigning secondary structure to the amino acids from the atomic coordinates of the protein's structure, the DSSP algorithm [207] is considered as the standard method. The DSSP database contains an associated entry for each protein structure in the PDB database, which includes information on the exposure of each residue automatically inferred from the 3D structure. This information is available under the column "ACC" (figure 5.2).



```
 0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0    LADDERS PER SHEET            .
#  RESIDUE AA STRUCTURE BP1 BP2  ACC     N-H-->O    O-->H-N    N-H-->O    O-->H-N    TCO  KAPPA ALPHA  PHI   PSI    X-CA   Y-CA   Z-CA
   1    1 A M              0   0  207      0, 0.0    86,-1.0     0, 0.0     2,-0.3   0.000 360.0 360.0 360.0-114.0   77.8    1.1   21.0
   2    2 A Q  B     -a   87   0A 129     84,-0.2    86,-0.2    86,-0.1     2,-0.2  -0.898 360.0-121.0-140.0 167.4   74.5    3.1   21.1
   3    3 A L        -     0   0    9     84,-2.2     3,-0.1    -2,-0.3    86,-0.0  -0.452  51.0 -69.0-101.0 177.5   72.4    5.3   23.4
   4    4 A K        -     0   0   86    136,-0.2    -1,-0.1    -2,-0.2   143,-0.1  -0.461  58.4-104.3 -71.0 136.5   68.8    4.8   24.5
   5    5 A P        -     0   0   50      0, 0.0     2,-0.3     0, 0.0    -1,-0.1  -0.308  40.6-158.7 -60.5 145.1   66.3    5.3   21.7
   6    6 A M        -     0   0   24     82,-0.1     2,-0.5    -3,-0.1    85,-0.1  -0.968  19.8-118.4-131.0 145.0   64.4    8.6   22.0
   7    7 A E        -     0   0   70     -2,-0.3     2,-1.0   140,-0.3     5,-0.1  -0.689  31.1-135.0 -80.3 124.3   61.1    9.8   20.6
   8    8 A I        +     0   0   32     -2,-0.5    44,-0.1     4,-0.1    -1,-0.0  -0.741  53.6 118.3 -89.4 104.5   61.8   12.8   18.3
   9    9 A N  S  > S-     0   0   35     -2,-1.0     4,-2.5     1,-0.1     5,-0.3  -0.958  75.4 -94.5-153.7 163.0   59.3   15.6   19.0
  10   10 A P  H  > S+     0   0   16      0, 0.0     4,-2.9     0, 0.0    18,-0.4   0.871 121.2  51.0 -49.8 -44.3   59.5   19.3   20.2
  11   11 A E  H  > S+     0   0  123      2,-0.2     4,-1.7     1,-0.2     5,-0.1   0.933 111.7  44.9 -62.8 -48.5   58.9   18.3   23.8
  12   12 A M  H  > S+     0   0    7      2,-0.2     4,-1.4     1,-0.2    -1,-0.2   0.939 117.3  45.4 -62.1 -46.3   61.6   15.7   24.0
  13   13 A L  H  X S+     0   0    0     -4,-2.5     4,-2.4     1,-0.2     3,-0.3   0.914 110.4  53.6 -63.8 -43.2   64.1   17.9   22.2
  14   14 A N  H  X S+     0   0   28     -4,-2.9     4,-2.0    -5,-0.3    -1,-0.2   0.829 105.8  54.1 -62.1 -30.5   63.2   21.0   24.3
  15   15 A K  H  X S+     0   0   70     -4,-1.7     4,-2.0    -5,-0.2    -1,-0.3   0.863 108.5  49.9 -71.0 -32.1   63.9   18.9   27.4
  16   16 A V  H  X S+     0   0    1     -4,-1.4     4,-1.4    -3,-0.3    -2,-0.2   0.890 107.0  53.3 -70.8 -40.5   67.3   18.1   25.9
  17   17 A L  H ><>S+     0   0    0     -4,-2.4     5,-2.2     1,-0.2     3,-0.6   0.951 110.4  49.4 -57.4 -47.2   68.0   21.8   25.3
```

Figure 5.2: An example of a DSSP file. The highlighted yellow column is for amino acids, accessibility values are colored in green.

Using the entries from the DSSP database, the values of relative accessibility to the solvent were calculated for each of the residues of every selected protein. To calculate these relative accessibility values, the ACC (accessibility) value (from DSSP) is normalized by the maximum residue accessibility for each of the 20 amino acids as defined by [208]. Thus the relative accessibility is calculated as

$$Relative\ Accessibility = \frac{ACC}{Maximal\ Residue\ Accessibility}$$

A relative accessibility value of 1 means high accessibility. This means the residue is exposed to the solvent. Similarly, a value of zero means no accessibility, which indicates the residue is buried in the protein structure. An example protein is shown in figure 5.3 displaying three different levels of relative accessibility. Around 50% of all the residues of the proteins considered had a relative accessibility below 0.1, with 32% above 0.5 and only 10% above 0.9, but these values depend very much on the type of amino acid considered.

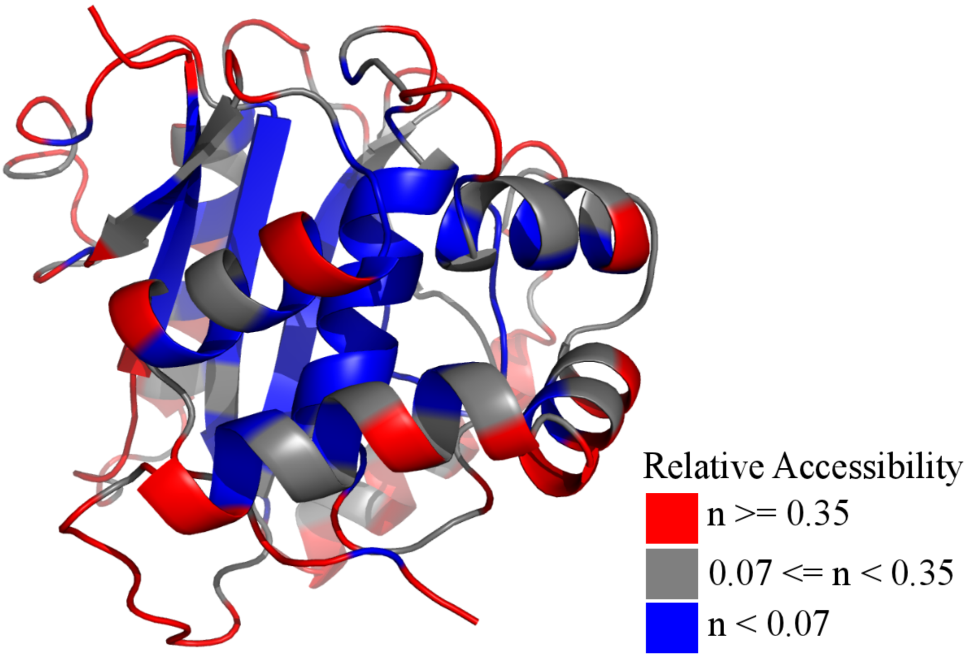


FIGURE 5.3: Visualization of DSSP ranges on protein structure. The protein structure is colored according to the relative accessibility values.

Next the distribution of relative accessibility values for each of the 20 amino acid types is calculated. For this purpose, the relative accessibility range (0 to 1) is divided into 10 equal sized bins and the number of residues falling in each bin is counted. This procedure is followed for each of the 20 residue types (figure 5.4).

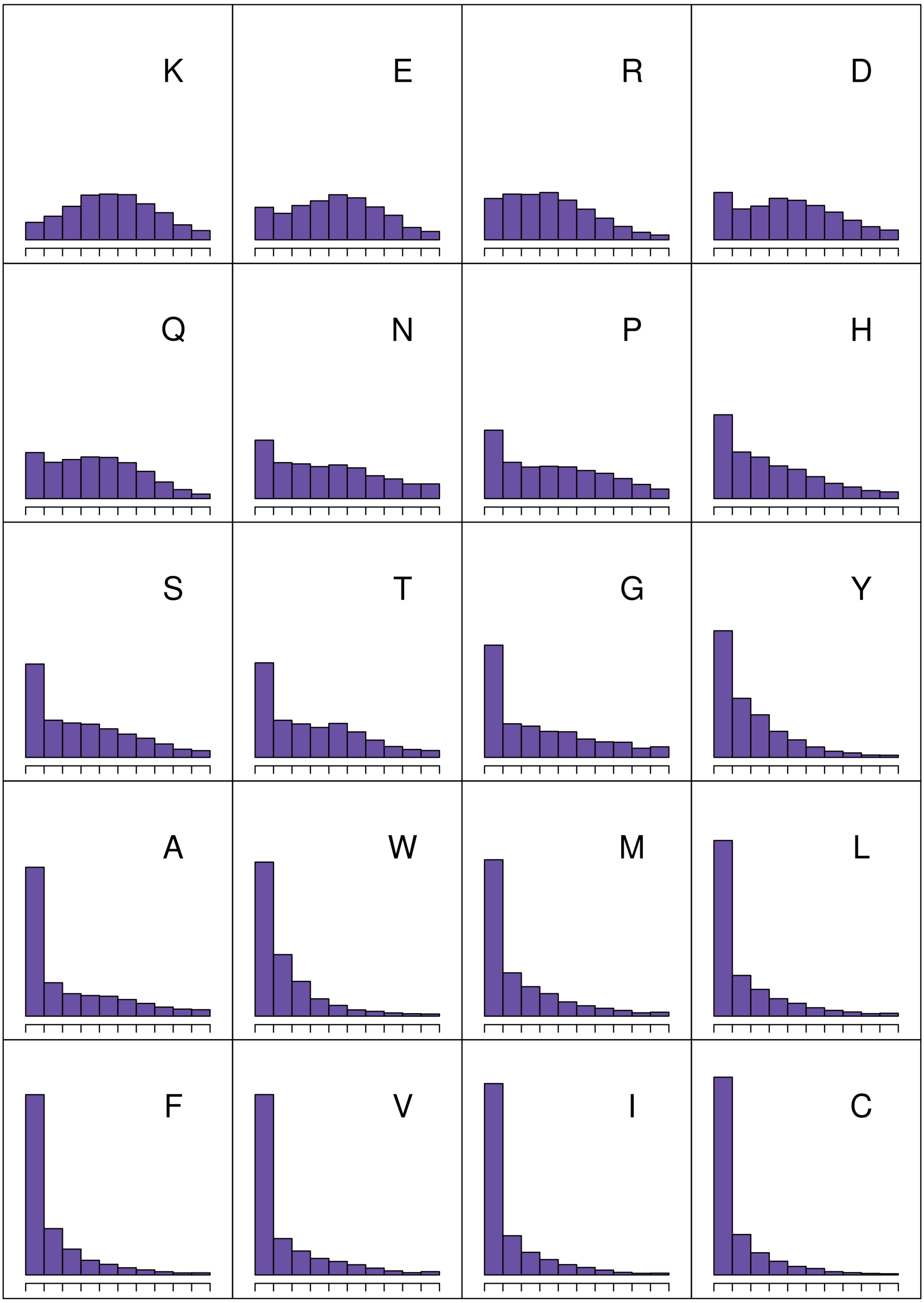


FIGURE 5.4: Residue exposure frequency distributions (from buried to exposed) for each of the 20 amino acids in the proteins of known structure and experimentally verified location used to train the algorithm. The graphs were arranged according to similarity.

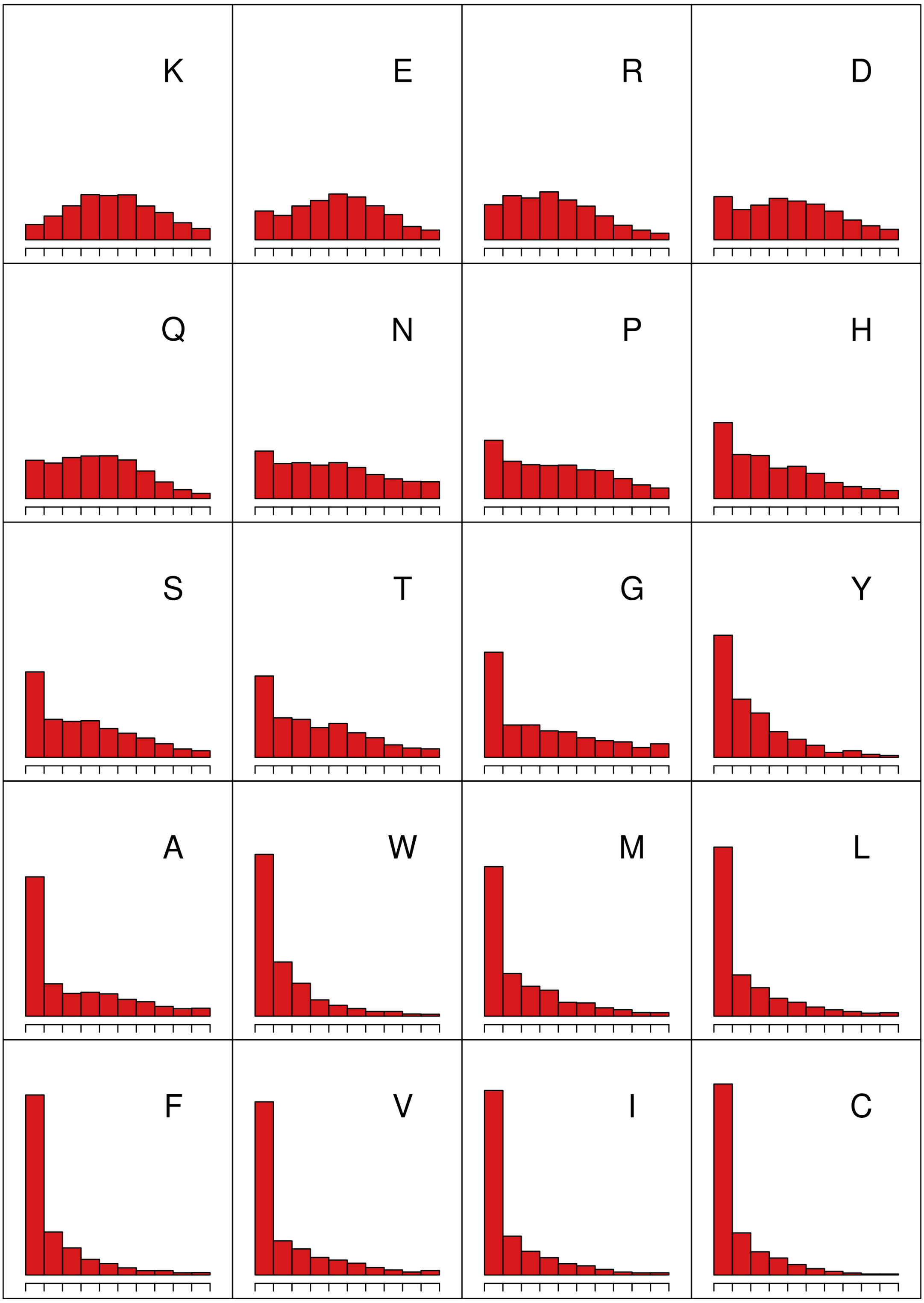


FIGURE 5.5: Residue exposure frequency distributions (from buried to exposed) for each of the 20 amino acids for the dataset of nuclear proteins. The graphs were arranged according to similarity.

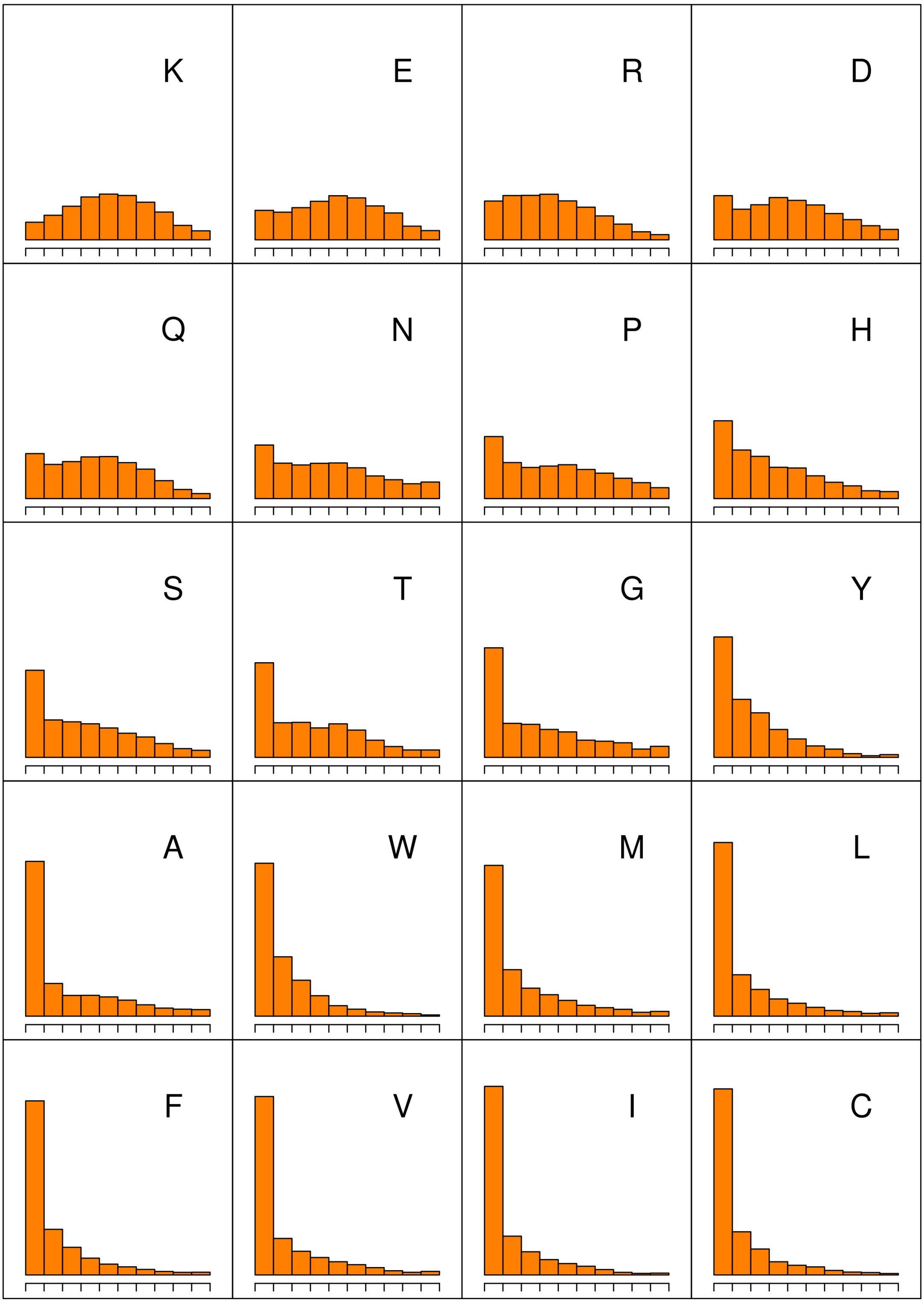


FIGURE 5.6: Residue exposure frequency distributions (from buried to exposed) for each of the 20 amino acids for the dataset of nucleocytoplasmic proteins. The graphs were arranged according to similarity.

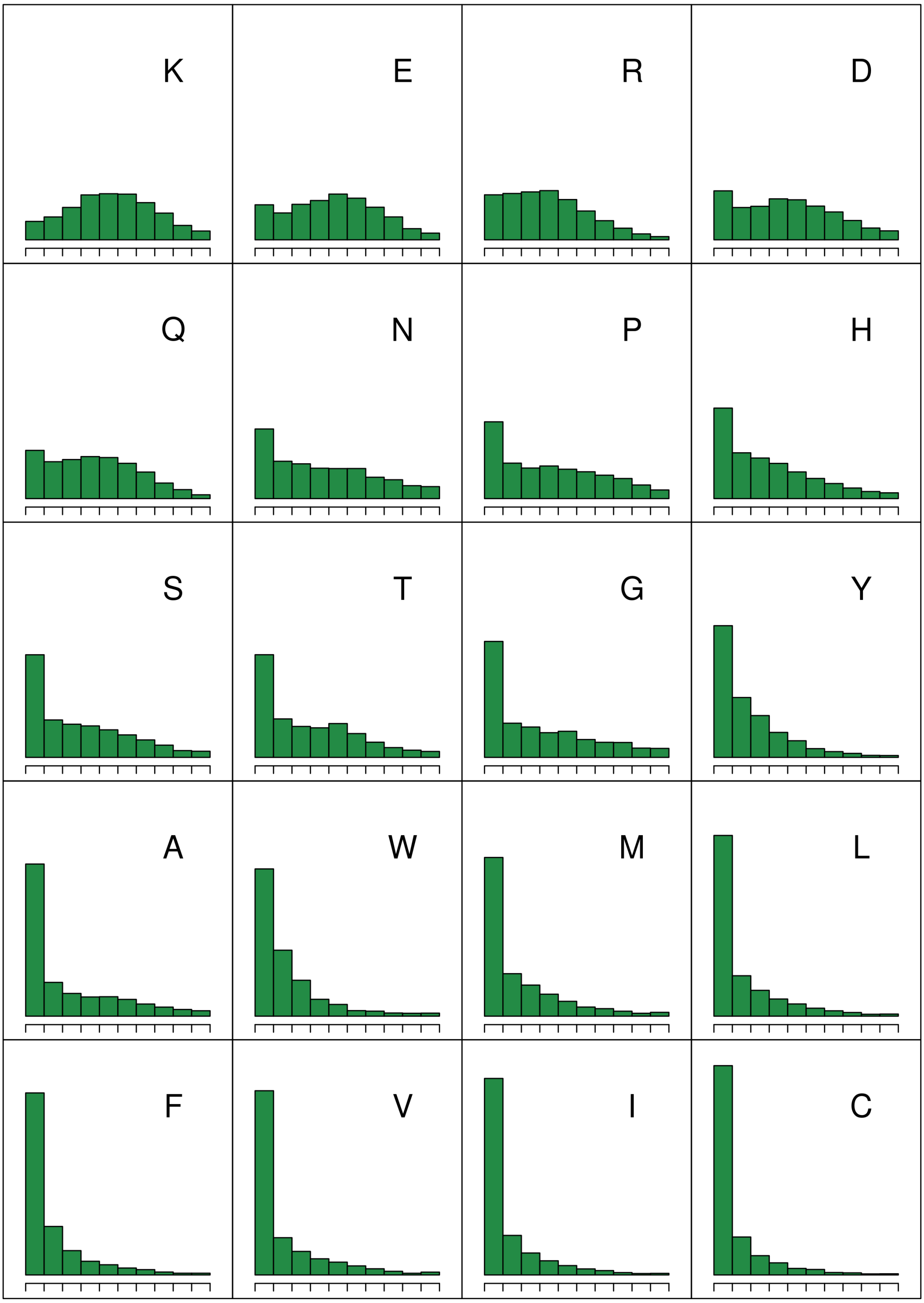


FIGURE 5.7: Residue exposure frequency distributions (from buried to exposed) for each of the 20 amino acids for the dataset of cytoplasmic proteins. The graphs were arranged according to similarity.

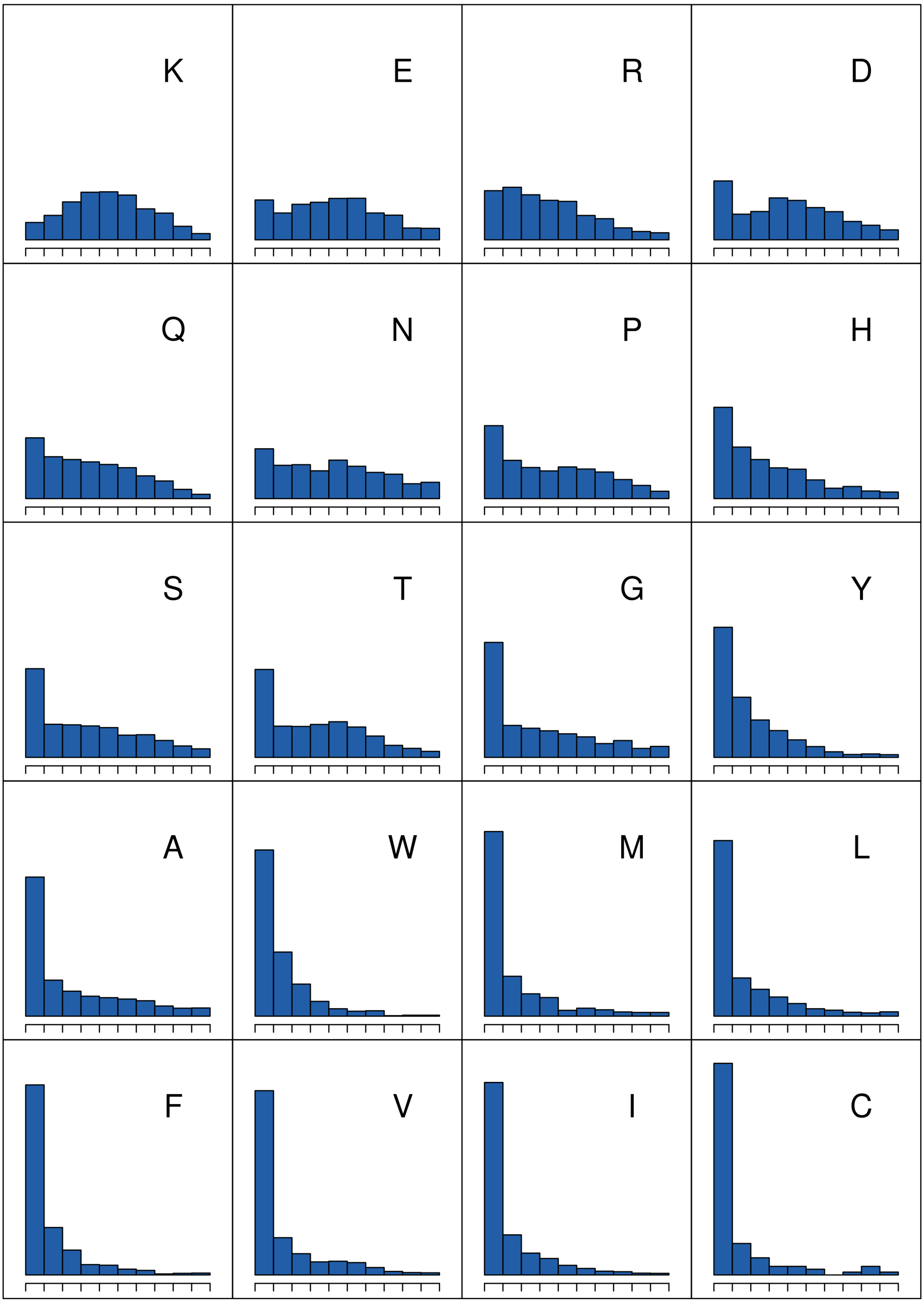


FIGURE 5.8: Residue exposure frequency distributions (from buried to exposed) for each of the 20 amino acids for the dataset of extracellular proteins. The graphs were arranged according to similarity.

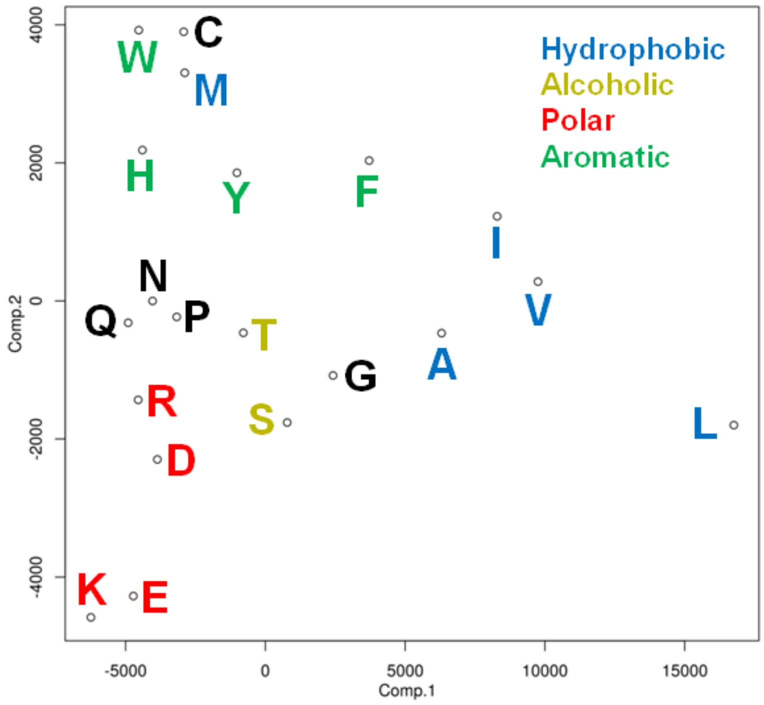


FIGURE 5.9: Principal component analysis of the vectors of exposure of the 20 amino acids shown in figure 5.4. Amino acids with similar properties appear close in the projection: polar residues like arginine (R), aspartic acid (D), glutamic acid (E) and lysine (K) group together. Same is true for alcoholic residues such as threonine (T) and serine (S), and for aromatic residues (tryptophan (W), histidine (H), tyrosine (Y), phenylalanine (F)).

This reflects the similarities between amino acid chain properties so that, for example, the distributions of hydrophobic residues are similar to each other. Accordingly, we observed that residues with side chains belonging to the same physicochemical property group show similar frequency distributions (figure 5.4). For example the hydrophobic residues isoleucine (I), valine (V), leucine (L) and alanine (A) show very similar distributions with a very high frequency in the low accessibility region and fewer residues in the high relative accessibility region. Principal component analysis (PCA) of these data shows this more prominently (figure 5.9).

The distribution of exposure values for the 20 different amino acids is also calculated for each of the four protein classes separately. Through the comparison of the distribution of class specific amino acid exposure values we observed variation for particular amino acids and protein locations. For example, when we compare the distribution of exposure values for glutamine (Q) in different location classes we can see that glutamines in extracellular proteins are more buried than in intracellular proteins. Conversely, cysteines in extracellular proteins have a distinct peak at high exposure values, which is absent in intracellular proteins. These differences imply that exposure values can be used to predict protein location.

5.4 Discussion

In this chapter, we presented a detailed analysis of amino acid composition and its exposure variation with the subcellular location of the corresponding protein. Our study demonstrated that the residue exposure in a protein varies with its subcellular location. This also implies that the residue exposure values can be used, in combination with amino acid composition, for the prediction of subcellular location of proteins. We selected proteins for our analysis with experimentally verified subcellular location. Although it is possible that the location annotation of a protein in UniProt may contain errors, it is very unlikely that such errors affect the proteins used in this study because we used only proteins that have a three dimensional structure available in PDB and usually those are well studied and characterized. The residue accessibility values of the amino acids of these proteins were calculated using the DSSP database. The tool NACESS [209] could have been also applied for this purpose, although we found no significant difference in relative accessibility and residue exposure frequency for many randomly selected proteins.

The calculated vectors of accessibility distribution for each amino acid reflect their physicochemical properties. We used principal component analysis (PCA) to represent this high-dimensional data in a low-dimensional form. PCA is a robust method for dimensionality reduction without a serious loss of information. In the projection of the vectors of accessibility distribution for the 20 amino acids, residues with similar properties occupy close positions. The analysis also indicates that more than pKa (acidity) or net charge, polarity and size drive residue exposure.

Chapter 6

Development of the NYCE Algorithm

6.1 Motivation

Proteins from different subcellular locations have distinct residue exposure patterns. The detailed analysis of proteins from four different classes, described in chapter 5, establishes this fact and suggests that residue exposure patterns can be used for subcellular location prediction. An example case, shown in figure 6.1, represents the distribution of glutamine (Q) residue exposure values in four different location classes. We can clearly see that glutamines in extracellular proteins are more buried than in intracellular proteins. Even among the intracellular location classes, there are coherent differences in the protein residue exposure patterns. Compared to cytoplasmic proteins, the glutamine residues are more exposed in nuclear proteins.

The general principle that a protein's location is correlated to its residue exposure properties has been studied earlier by Andrade et al. [131], though the systematic study of the relation between residue exposure and subcellular location is lacking. Here we first analysed the different ranges of exposure values and tried to find out which range contributes most to protein location. For this purpose, the relative

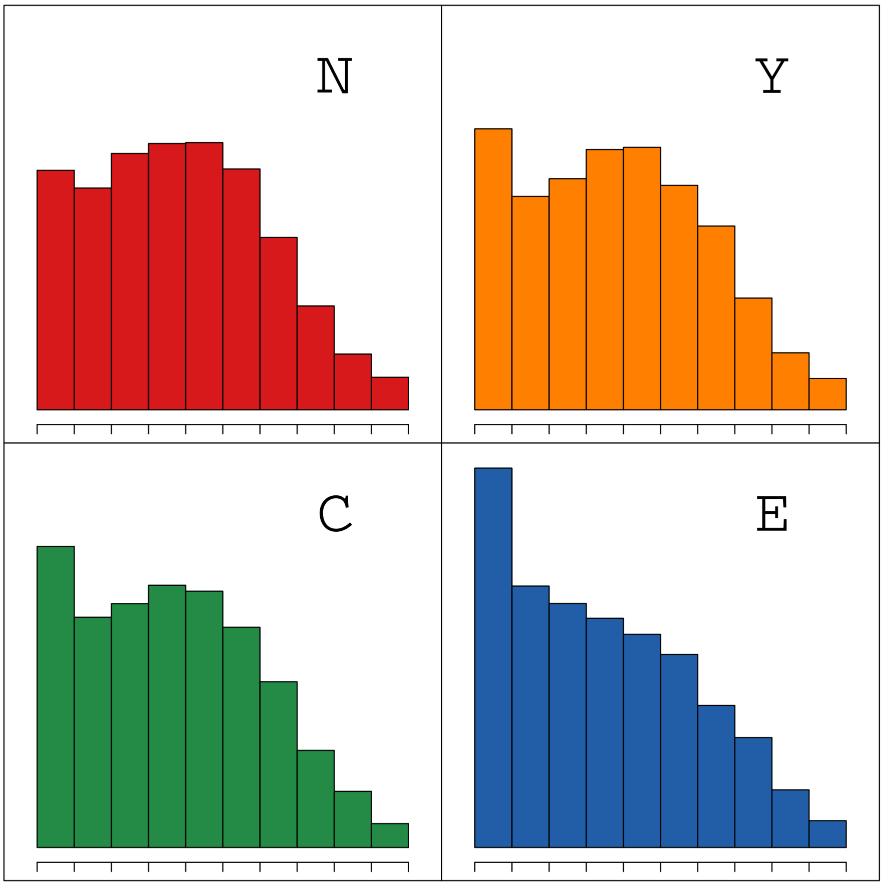


Figure 6.1: Distribution of values of exposure of glutamine (Q) in different location class proteins: nuclear (N), nucleocytoplasmic (Y), cytoplasmic (C), and extracellular (E).

accessibility value is divided into six ranges, making sure that an equal number of residues fall in each range. Using SVMs we carried out the location wise study on the effect of residue exposure range on classification accuracy. This analysis provided us insight into the amount of information each exposure range has. To apply these newly discovered principles in practical use, we used a two level classification approach. For the final classification we developed a hybrid method that combines SVMs and an ANN, trained on proteins of known location and structure for the prediction of the four locations mentioned above: nuclear (N), nucleocytoplasmic (Y), cytoplasmic (C) and extracellular (E). We also adapted the method for use on sequences of unknown structure by using predicted amino acid exposure values with reasonable performance. To provide a convenient tool for location analysis from protein sequence only, we also implemented the algorithm in form of a web tool.

Range	DSSP	SABLE
1	[0,0.01]	0
2	[0.01, 0.08]	1
3	[0.08, 0.21]	2
4	[0.21, 0.37]	3
5	[0.37, 0.57]	4
6	[0.57, 1.00]	[5, 9]

Table 6.1: Ranges of exposure used and their corresponding DSSP and SABLE values

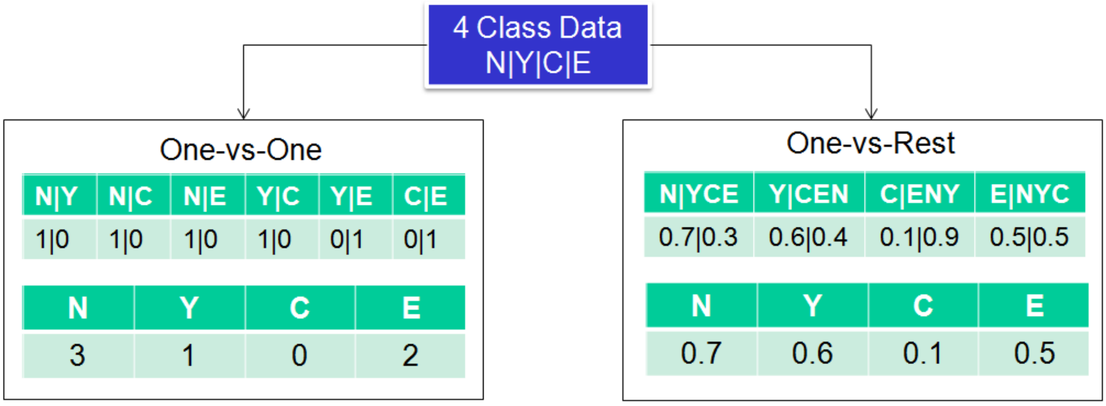
6.2 Calculating amino acid composition vectors

From the normalized relative accessibility of each protein, 20 and 40 component vectors are computed. The amino acid composition vector of a protein is a vector of 20 components, one for each amino acid type. Each component i is the fraction of residues of type i in the protein. Therefore the sum of the components is equal to one. The amino acid composition vectors at six ranges of residue exposure values were computed such that at every range there is an almost equal number of residues (Table 6.1).

This allows us to compare and combine different ranges in terms of power for prediction of protein location. For particular calculations, the 40-component vectors were created by combining two 20-component vectors.

6.3 First step classification using SVMs

At the first step of classification, an SVM learning method is applied. For this purpose the software library LIBSVM, Version 3.11 [185] is used. As discussed in the previous chapter, SVMs are a binary classification algorithm. For the protein location classification problem, the extension of SVMs to multiclass data is required. There are two main approaches to solve the multiclass (N-class) problem: one-vs.-one approach or one-vs.-rest approach. To solve N-class problems the one-vs.-one approach creates $N \times (N-1)/2$ binary classification models. In the next step it applies majority voting for a final decision. The one-vs.-rest approach creates only N different models and a final decision is based on maximum probability. This is also described as "winner takes all" approach.



N|Y	N|C	N|E	Y|C	Y|E	C|E
1|0	1|0	1|0	1|0	0|1	0|1

N	Y	C	E
3	1	0	2

N|YCE	Y|CEN	C|ENY	E|NYC
0.7|0.3	0.6|0.4	0.1|0.9	0.5|0.5

N	Y	C	E
0.7	0.6	0.1	0.5

FIGURE 6.2: SVM based multiclass classification approaches. Two possible approaches, One-vs-One and One-vs-Rest are represented along with the required number of binary classification models.

To decide which of these classification strategies should be used, it is important to contemplate the nature of the classification problem in hand [210]. Consider the case of a protein that is localized in the nucleus. Classifying this protein using the one-vs.-one approach will require $(4 \times (4 - 1))/2 = 6$ binary classification models. Out of these 6 classification models, only 3 will have the option to classify the protein in the correct class "nuclear". The other 3 classifiers will necessarily classify the protein in a category other than nuclear, therefore wrong. The one-vs.-rest approach will use only 4 classification models. Out of these 4 classification models, 1 classifier will have the option to classify the protein in the nuclear class while the remaining 3 classifiers might correctly classify the protein in the 'rest' category. Considering this fact we chose the one-vs.-rest approach for multiclass classification. For each exposure range and their combinations, we trained 4 one-vs.-rest SVM models.

6.3.1 Data balancing and training

Vectors of amino acid composition for a set of proteins of known structure and location using amino acids in different ranges of exposure were used as input data for LIBSVM. The dataset in-hand is highly unbalanced (Table 5.1). In an unbalanced dataset, where one class instance far outnumbers other class instances, SVMs perform poorly and can produce biased results. For instance, if a classifier classifies a data set where the class ratio is 3:1, the classifier can perform at 75% accuracy just by classifying all data-points in the larger class. To overcome this problem the data-balancing method is applied. For each of the four location classes (N, Y, C and E) one was taken as positive and an equal sized negative dataset was created with members from the other three classes. When possible, the negative dataset contained the same amount of sequences for each of the 3 classes. When using C as the positive set (543 sequences) there were not enough E proteins to be used as negatives ($123 < 534/3 = 181$). In this case all E proteins were used as negatives and an equally sized set was taken from Y and N proteins to complete the negative set (210 from each). For each SVM training a 10-fold cross validation was performed. For this purpose the data was randomly divided into 10 sets. For each of the 10 cross validations one set was used as test data and the others were used as training data. To obtain an optimized SVM model the parameter space of the SVM was searched. The parameter values that produce the best accuracy were recorded and used for the optimized model. As the training datasets were balanced it was safe to use accuracy as performance measure. The accuracy of the SVM was evaluated as the fraction of proteins in the test set correctly predicted. An average accuracy value was calculated from the 10-fold cross validation tests. Performance of different range vectors were compared using ROC (receiver operating characteristic) curves.

6.3.2 SVM classification using vectors of amino acid composition in selected ranges

As described in section 6.2, we separated the values of amino acid exposure in six percentiles (1-6, from buried to exposed). The vectors of amino acid composition for different combinations of these six ranges were tested. Initially we tried vectors with 20 components (one for each amino acid) describing the composition of residues found within a particular range of exposure values. For example, the range “1” composition vector for a protein would be defined by the distribution of amino acids of this protein with exposure values in the most buried category. The range “5 6” would be defined by the amino acids in the two most exposed categories. The range “1 2 3 4 5 6” would be the amino acid composition of the entire protein and so on. We then trained an SVM on such amino acid composition vectors for proteins from each of the four location categories. The accuracy of the classifier was distinctively better for extracellular proteins and worst for nucleocytoplasmic proteins (figure 6.3).

Interestingly, for nuclear proteins, and less so for nucleocytoplasmic and cytoplasmic proteins, the middle ranges of exposure (3 and 4) seem to contain less signal about the location of the protein. For extracellular proteins, buried residues contain more information on the location of the protein than exposed residues. In any case, the complete protein amino acid composition (full range: 1 2 3 4 5 6) was a better predictor than each of the six individual ranges, with composition from multiple ranges, e.g. (1 2), (3 4 5 6), close. The bad performance of vectors of residues in smaller ranges may be due to the fact that we are dealing with proteins with an average size of 322 amino acids and the resulting range-specific amino acid composition vectors may be based on small numbers of amino acids. This effect is obviously reduced when the full range or a combination of ranges is used. Since combined ranges seemed to perform next to full-range we wondered if combining these vectors could outperform full-range vectors. Therefore, we next tested SVM classifications using as training 40-component vectors that combined two different 20-component vectors. In particular, the 40-component vector combining the 20-component vectors for residue composition in the three most buried categories with the 20-component vectors for residue composition in the three most exposed categories (1 2 3, 4 5 6) provided on average better predictions than the full-range vector for the four location categories (figure 6.3). Generally, this vector produced better results than other combinations excluding some ranges (e.g. (1 2, 5 6)) or using scrambled residue ranges (e.g. (1 3 5, 2 4 6)).

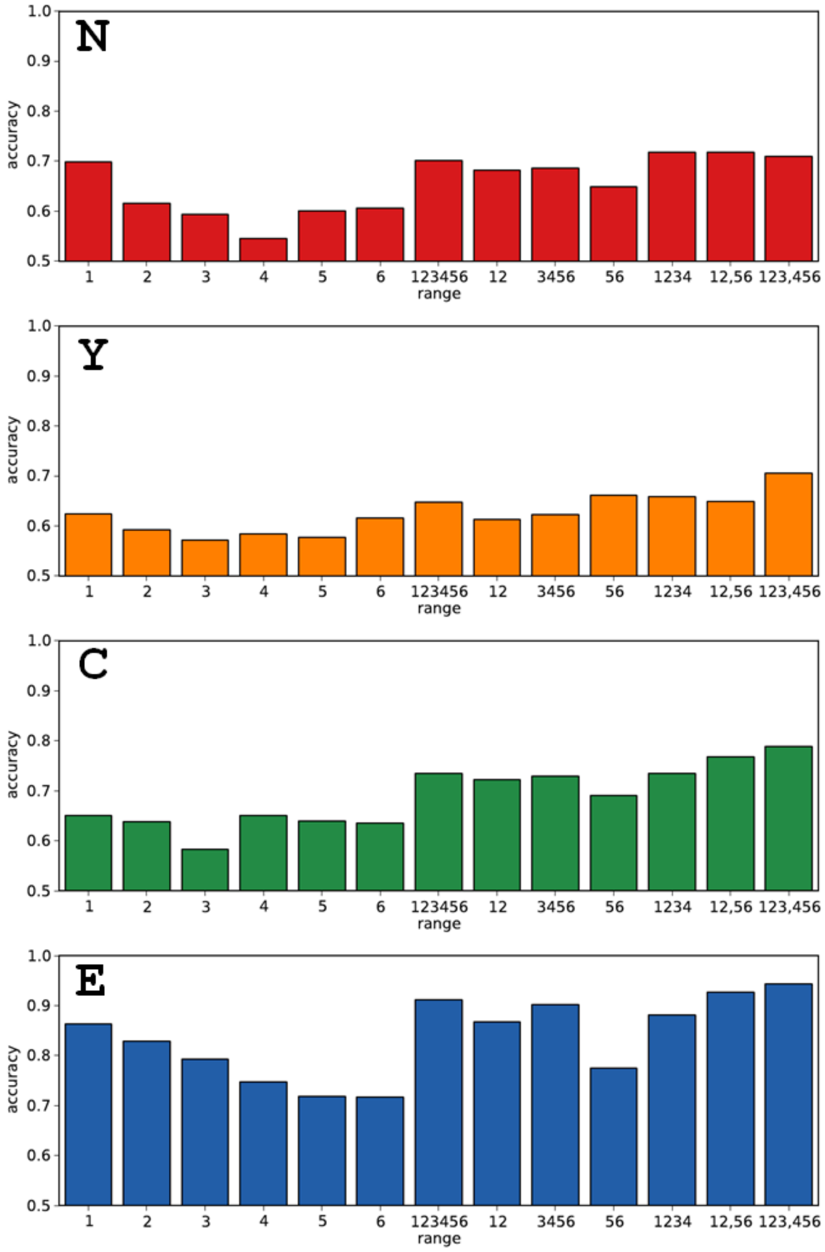


FIGURE 6.3: Accuracy of one-vs.-rest SVM classifications for nuclear (N), nucleocytoplasmic (Y), cytoplasmic (C) and extracellular (E) proteins using residues in different ranges of exposure (1-6, from buried to exposed).

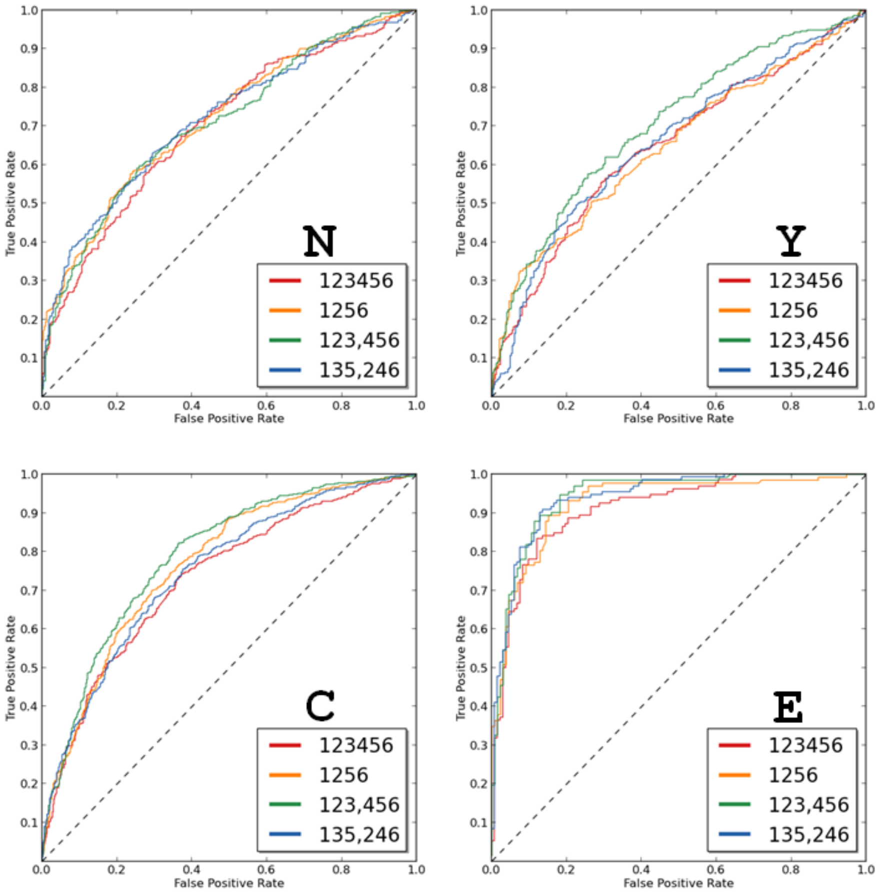


Figure 6.4: ROC curves of one-vs.-rest SVM classification for four location classes using composition vectors of residues in different ranges: 20-component vector based classification (ranges 123456 and 1256) and 40-component vector based classification (ranges 123,456 and 135,246).

Since from each one-vs.-rest SVM model we obtain a probability of being in a location class, it is possible to evaluate the accuracy of the model using a threshold for this probability. That is, we can compute the recall and precision of the predictions above various cut-offs of probability. The plot of these values as ROC curves confirms that the extracellular class is predicted the best and that the 40-component vector (1 2 3, 4 5 6) provides better predictive power than full composition (figure 6.4). To rule out the possibility that the superiority of the 40-component vector would be due to the higher amount of components, we tested a 40-component vector with scrambled ranges (1 3 5, 2 4 6), which performed poorly (figure 6.4).

To combine multiple SVM predictions into a single one we applied a simple "winner-takes-all" strategy, that is, the prediction with the best score is selected.

ROC curves indicated that the 40-component vector (1, 2 3 4 5 6) performed best against other 40-component vectors (e.g. (1 2 3, 4 5 6)) or the full range 20-component vector (1 2 3 4 5 6) (figure 6.6). We then applied the "winner-takes-all" strategy to the three SVM sets mentioned above (that is, a set of 12 SVMs), but this did not improve performance significantly (dotted cyan curve in figure 6.6).

6.4 Combining class probabilities with an ANN

We wondered if combining SVM scores for different locations and ranges using an Artificial Neural Network (ANN) for a second level of classification, as opposed to just taking the best score prediction, could improve the accuracy of the method. The ANN used for this purpose is a multilayer perceptron, consisting of 3 layers: input, hidden and output layer. The output probability values from one-vs.-rest SVM models are used as input for the ANN. The number of neurons in the hidden layer was optimized for maximum accuracy, as well as the type and number of SVMs using as input (figure 6.5). For each individual range, the probability values obtained from 4 different one-vs-rest SVM models are fed into the ANN. Thus, for such cases the input layer of the network consists of 4 input neurons taking the probability values from one-vs.-rest SVM models for each of the four categories as input. For better accuracy the combinations of ranges are also analysed using an ANN. Thus for combination of 2 and 3 ranges, the ANN input layer consists of 8 and 12 input neurons, respectively.

For all the cases the output layer of the neural network consists of 4 neurons, one for each location class. The back-propagation algorithm is used for training and the number of neurons in the hidden layer was optimized by 10-fold cross-validation. Different combinations of SVM models trained with different range vectors were tested for better accuracy. The best result was obtained for 28 hidden-layer neurons and 12 input-layer neurons; the inputs were obtained from four SVMs using 40-component vectors for ranges 1 2 3 and 4 5 6, four SVMs using 40-component vectors for ranges 1 and 2 3 4 5 6, and four SVMs using 20-component

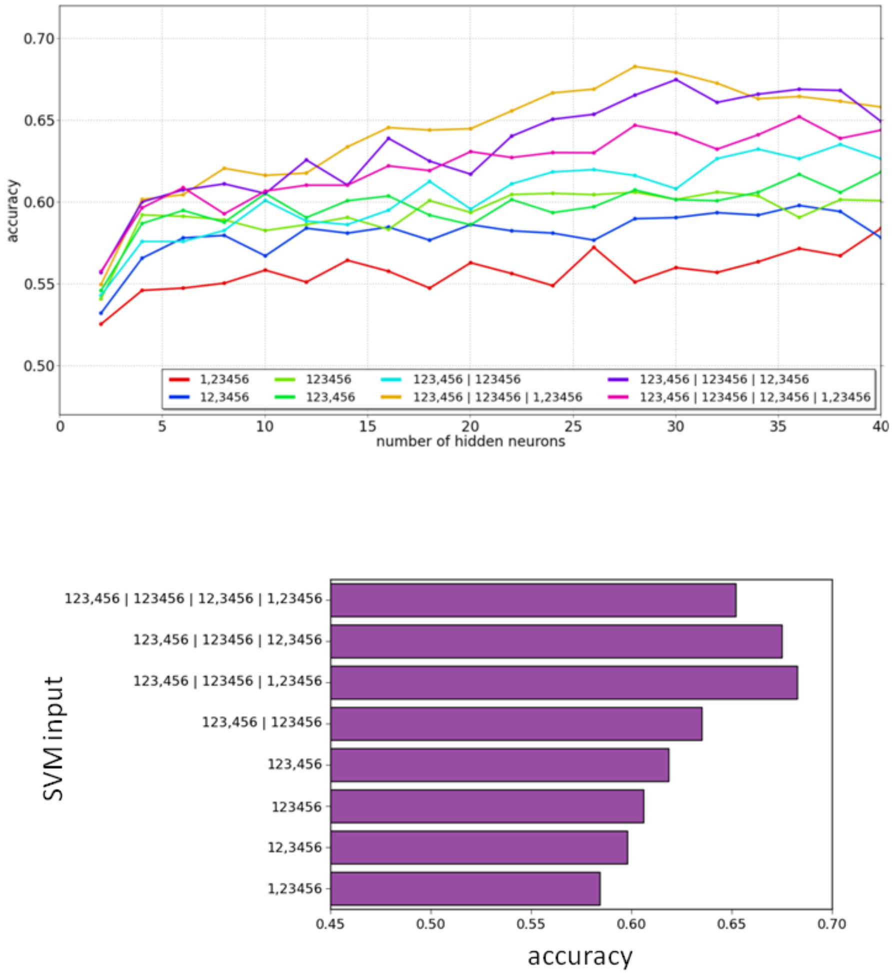


Figure 6.5: Optimization of the artificial neural network (ANN). (Top) ANNs were optimized using different numbers of hidden neurons and SVM types. (Bottom) Best accuracy value obtained. The legends indicate the type of SVM input used. SVM ranges and vectors (of 20 or 40 components) are indicated as in figure 6.4. Use of multiple sets of SVMs are indicated by labels using "|" as separator. For example, the best accuracy value (0.68) was obtained using as input three sets of SVMs, two of them trained on 40-component vectors, and another trained on 20-component vectors.

vectors of full protein composition (accuracy 68%; figure 6.5). Increasing the number of SVMs used as input eventually decreased accuracy, probably due to over-training of the ANN. The final number of connections in the optimal ANN, $(12 \times 28) + (28 \times 4) = 448$, is well below the number of examples used for the training (1,358).

Like we did for SVM optimization, the performance of different ANN models were compared using average accuracy and ROC curves. The output of ANN models

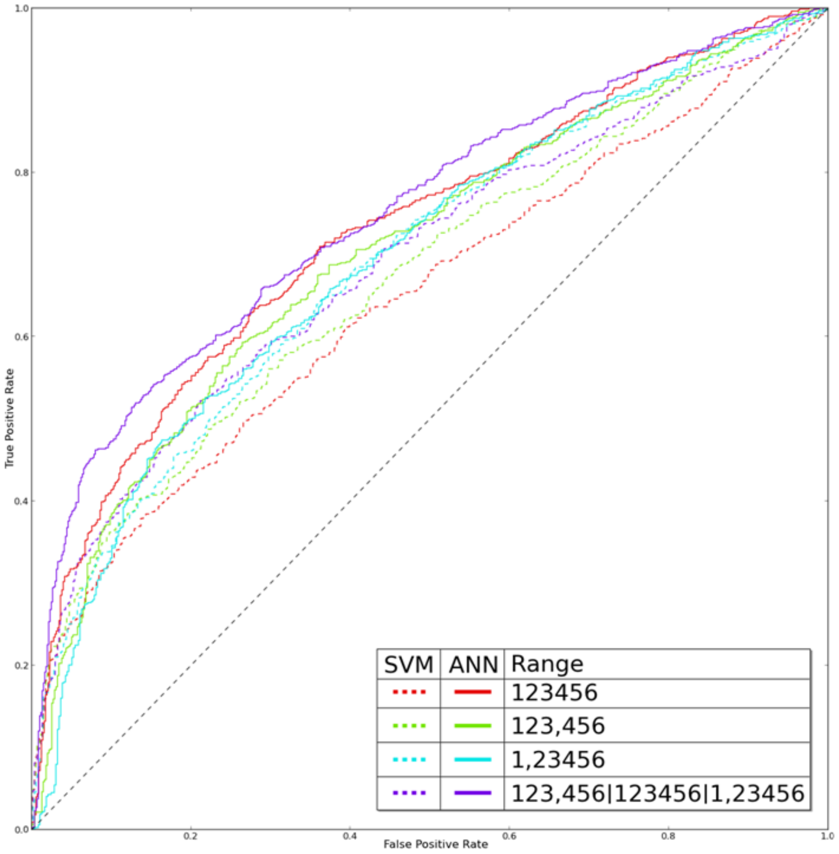


FIGURE 6.6: ROC curves from SVM classifications (winner-takes-all strategy) and ANN classifications that use as input the SVM values. For SVMs the ROC curves (dotted lines) were made by taking the best prediction from sets of SVMs (winner-takes-all strategy). Either best of four SVMs for each location category (red, green and cyan dotted curves indicating the different ranges), or best of 12 SVMs (the combination of three SVM types is indicated with pipe signs indicating the vectors; violet dotted curve) used. For ANNs the ROC curves (continuous lines) used just the ANN output.

was also compared against the SVM models. ROC curves for the ANN classifications indicate that they improve the predictions over the SVMs used as input, and confirm that the ANN selected performs best (figure 6.6). This combination of SVM inputs and ANN architecture was therefore selected for further work and finally for implementation as a public tool.

6.5 Prediction of location for proteins without structural information

Our next goal is to apply the predictive architecture optimized above to protein sequences. Our method uses as input the composition of residues of a protein in six different ranges of exposure. However, generally a given protein sequence has no 3D-information and therefore no known exposure values. Thus, we first need a method to provide predicted exposure values for the residues in the protein sequence whose localization has to be predicted.

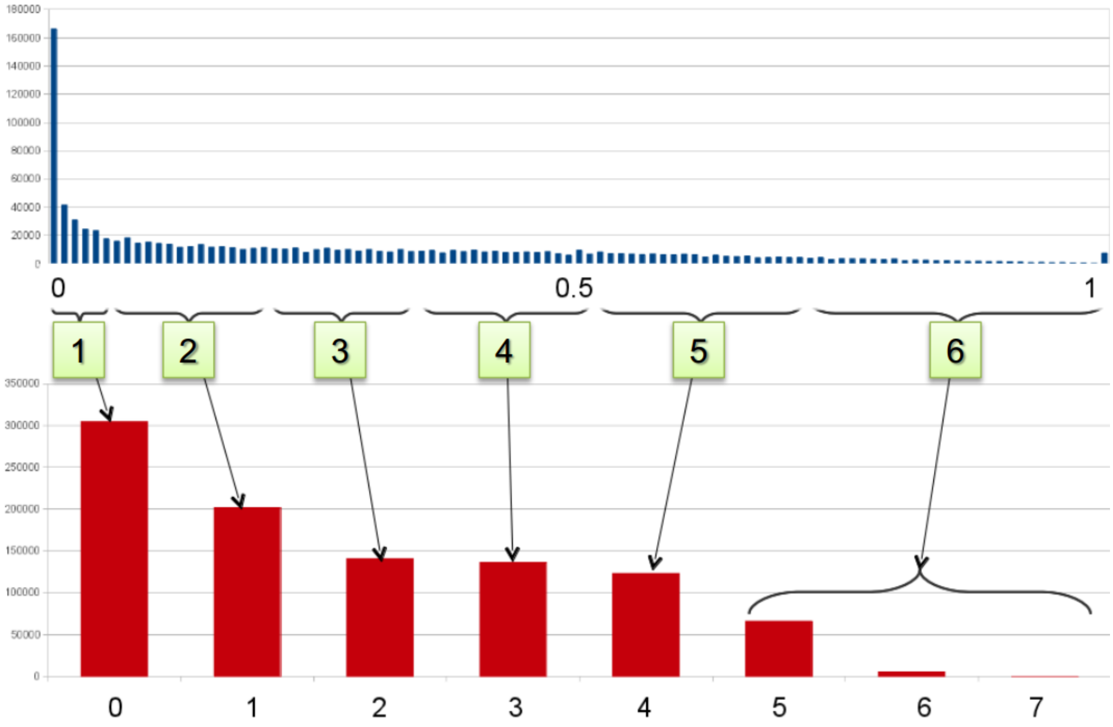


FIGURE 6.7: Mapping of DSSP and SABLE scores. The 6 ranges of distributions from the DSSP are mapped to the distribution of 9 possible SABLE values, according to percentile distribution as well as possible.

To obtain predicted exposure values alone from sequence we have used a method called 'SABLE', which predicts exposure based on residue type and similarity to other sequences with high reliability [211, 212]. This tool predicts relative solvent accessibility of an amino acid residue on a scale from 0 to 9 with an approximate accuracy of 78%. In principle, the scoring scale of SABLE does not necessarily correspond directly to the scale of values of exposure that we obtained from proteins of known structure. Since the final model of NYCE is based on residues classified in six ranges of relative solvent accessibility values derived from DSSP it is required to map the SABLE predicted solvent accessibility values to those 6 ranges. For this purpose, the protein sequences with PDB information that were used to train the method were analysed by SABLE. The distribution of exposure values predicted by SABLE was compared against the DSSP based distribution. The 9 possible SABLE values were matched to 6 ranges according to percentile distribution as well as possible (Table 6.1). Finally, we equated SABLE scores 0 to 4 to our 3D-derived ranges 1 to 5, respectively, and the SABLE ranges of 5 and above (the less populated) to range 6, which was not perfect but approximated best the percentile distribution (Table 6.1, Figure 6.7). These values are then used to generate the different range exposure vectors derived from SABLE values that are fed into the classification model. The algorithm finally scores a protein for its membership to the four location classes. The accuracy of the predictions with the optimal architecture SVM-ANN method was of 62%, which, as it could be expected, was lower than the value of 68% obtained when using the obviously more accurate 3D-derived values.

6.6 Significance of the NYCE score

The algorithm of NYCE is trained exclusively on proteins from four locations. This poses the question of how will NYCE behave if the query protein sequence belongs to locations other than those considered in NYCE. To test this we ran the method on a set of 1358 eukaryotic proteins randomly selected from proteins with experimentally verified location but not assigned to nuclear, cytoplasmic or extracellular locations. To select the proteins, we followed a procedure similar to the one described earlier in section 5.2 and we refer to them as ‘other’ class.

All the proteins were analysed using NYCE and the score was recorded. We observed that more than 75% of these proteins not present in NYCE locations received scores below 0.4 (figure 6.8). This indicates that NYCE assigns lower

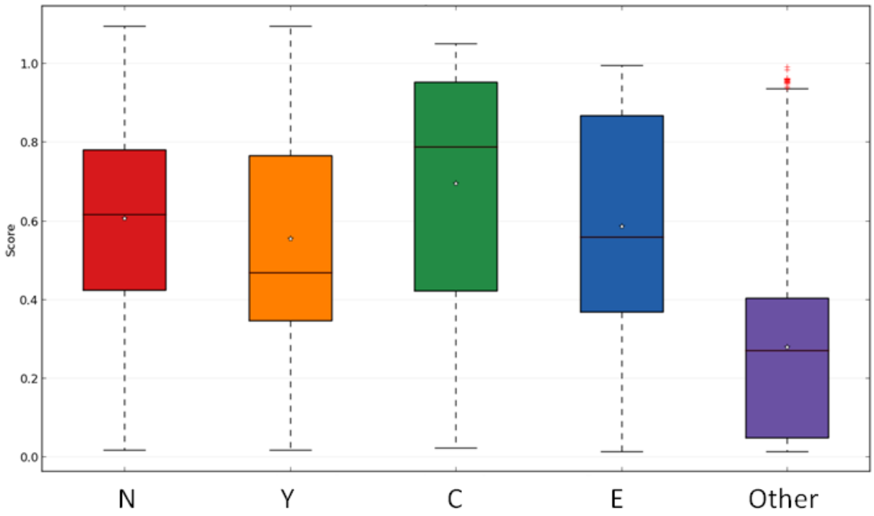


FIGURE 6.8: Box-plot of the scores obtained in the classification of proteins from four locations (nuclear (N), nucleocytoplasmic (Y), cytoplasmic (C) or extracellular (E)) or from other locations (Other). Proteins present in other locations received lower scores, indicating that the method can discriminate between them.

scores to the proteins that do not belong to one of the 4 classes considered in our study. Thus the NYCE score can be used for indicating the reliability of predictions and a NYCE score of 0.4 can be used as threshold. A NYCE score lower than 0.4 indicates that the protein might belong to some other location class. In the web tool, we provide all the 4 NYCE class scores, which can be more informative from the user's perspective.

6.7 Implementation of the NYCE web tool

We implemented the NYCE algorithm in a web interface that allows users to analyse the protein sequence of interest. The NYCE web tool is implemented via Django. Django is a free and open source high-level web application framework written in Python and follows the model-view-controller architectural pattern with an emphasis on the DRY (Don't Repeat Yourself) principle [213].

As an input, the web tool accepts a protein sequence in FASTA format and the corresponding residue exposure values of its amino acids as provided by SABLE. Using JavaScript, the input is first analysed to make sure that it has proper format

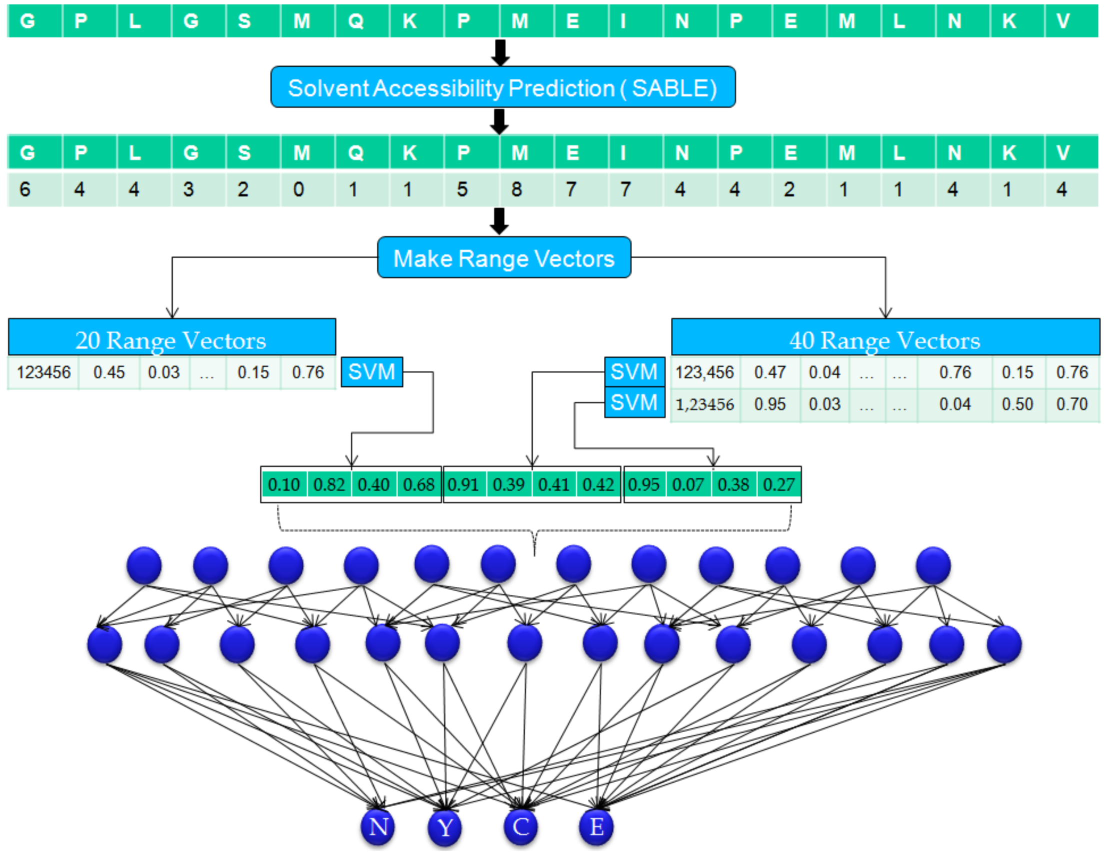


Figure 6.9: Schematic representation of the NYCE algorithm.

and length. For the NYCE web tool the input sequence must be at least 150 amino acids long. The residue accessibility values were first mapped to DSSP values. In the next step, NYCE calculates the frequency distribution of amino acids at different range values and makes 3 different range vectors (figure 6.9). Each range vector is fed to 4 different one-vs.-rest SVM models (one for each location class). The probability values obtained from these SVM models are combined by an optimized ANN model. The ANN gives a score for each of the 4 location classes. The web tool presents the score for each of the 4 location classes along with the input provided. The scores can be helpful for further interpretation of results. The web tool can be accessed at http://cbdm.mdc-berlin.de/amer/cgi-bin/nyce/ .

6.8 Discussion

In this chapter, we presented a novel approach for protein location prediction utilizing the residue exposure based signals. For the classification, different ranges of residue exposure values were analysed using a one-vs.-rest SVM approach. The probability values from the one-vs.-rest SVM model of best performing ranges were used as input for the ANN for the final classification. By comparing our two-step approach with SVM based winner-takes-all strategy, we showed that the two-step approach performs better. Furthermore, the comparison of scores obtained in the classification of proteins from four locations (nuclear, nucleocytoplasmic, cytoplasmic or extracellular) with proteins from other locations provides evidence that our method can differentiate between them.

On a technical note, our method illustrates how a multi-class problem can be approached by using a two-step approach where first SVMs of different types score class membership for each of the multiple classes and in a second step an artificial neural network (ANN) integrates the data and reassigns membership considering the scores from the SVMs. This type of two-stage mixed classifier might be especially useful in other situations where, as in our case, the number of examples to be used as test is relatively small and limits the size of the ANN that can be used without resulting in over-training. For example, we could not have trained the ANN directly on the 20 and 40-component vectors used as input for the SVMs with the few hundreds of examples of eukaryotic proteins of known location and structure available. In this respect, the SVM step can be considered as a kind of data compression prior to the use of an ANN. A Bayesian approach might also be feasible for this second step.

We note that our method depends on the quality of the predicted exposure values. Although SABLE has already high accuracy in the prediction of amino acid exposure [212], further developments in this field can be used to improve our predictions towards accuracy values close to those obtained when using 3D-derived values of amino acid exposure. Applying our approach to other protein location prediction problems, for example for prokaryotic proteins or for additional eukaryotic locations, is certainly possible but results will depend on the quality of experimental data on protein location and on the amount of signal for each location present in the sequences of experimentally verified locations.

Expanding the method will thus require careful selection of training datasets considering new taxonomic divisions and locations on a case by case basis. We expect that the development of novel techniques for high-throughput characterization of protein location might eventually facilitate such a development.

Chapter 7

Location Analysis for Paralog Protein Pairs

7.1 Motivation

One major challenge in protein location prediction is the prediction of location for homologous proteins. There can be very similar proteins that act in different subcellular locations. For example, the two well known protein-tyrosine kinases BMX and FRK are cytoplasmic and nucleo-cytoplasmic, respectively; however they both have two N-terminal domains (SH2-Protein kinase) that are responsible for about 25% sequence identity in a global alignment. Similarly, through a genome-wide comparison of protein subcellular location in *S. cerevisiae* and *S. pombe*, Yoshida et al. [59] have identified pairs of homologous proteins that do not have the same location. Homology is therefore not necessarily the best criterion to assign location to proteins.

The NYCE algorithm does not use protein homology and accordingly can assign different subcellular locations to homologous proteins. To test that NYCE can evaluate proteins independently of their homology, we analysed pairs of paralog proteins that are experimentally known to be in different subcellular locations. Our analysis shows that the performance of NYCE is significantly better compared to random tests where the pairs were assigned to each of four locations with equal probability. Furthermore, we compared the performance of NYCE with other state of the art subcellular location prediction methods. NYCE outperforms them both in terms of total accuracy and number of correctly predicted pairs of homologs. Thus the analysis verifies the fact that NYCE can find the appropriate location in cases where methods using homology would make a wrong inference.

7.2 Paralog selection

For the analysis, we collected pairs of homologous proteins with different experimentally known location (e.g. two homologous proteins where one is localized to the nucleus and the other to the cytoplasm). A relevant example of such a homologous protein pair (as mentioned earlier) are the tyrosine kinases BMX and FRK that share about 25% sequence identity and have similar domains. Despite such sequence homology, they have different subcellular location. To collect homologous protein pairs for the analysis, first we obtained pairs of homologous human proteins from the Eukaryotic Paralog Group Database [214]. These sequences are mapped to the corresponding Uniprot IDs. As discussed in chapter 5, shorter sequences can increase the noise. Thus, from the paralog protein pair data, we removed the protein pairs in which at least one protein sequence is shorter than 150 amino acids. For the remaining proteins in the dataset the subcellular location ontology terms in the UniProt records are analysed. From the data we selected the protein pairs such that both proteins in the pair have experimentally verified subcellular location information. Subsequently, we focused only on four locations: nuclear, cytoplasmic, nucleocytoplasmic and extracellular. For this purpose, all the proteins that have any location term other than these four were removed from the dataset along with the corresponding protein in the pair. Thus we are left with a set of paralog protein pairs where each protein is assigned to only one of the four considered location classes. From this set, we selected the pairs in which the proteins do not have the same subcellular location. The resulting paralog protein pair set contains 93 proteins that make up 64 paralog protein pairs. For simplicity we call it paralog-pair list.

7.3 Analysis of paralog protein pairs

Using NYCE, each protein pair in the paralog-pair list is analyzed in terms of its subcellular location. Out of 93 proteins, NYCE predicted the correct location for about 52% of the proteins, whereas 15 out of 64 pairs were predicted correctly. A pair's location is considered correctly predicted only if the location is predicted correctly for both proteins in the pair. In such a case of prediction for pairs, the simple notion of accuracy can be misleading.

It is possible that an algorithm achieves a high total accuracy in the prediction of individual sequences but shows poor performance at pair level. For example, consider 100 proteins that constitute 50 pairs. Assume we have a very efficient algorithm that predicts in total 80 proteins correctly, thus indicating a total accuracy of 80%. In such a case, depending upon the structure of pairs we may end up with a lower accuracy in the prediction of pairs; this can be as low as 60% if all 20 wrongly predicted proteins pair with correctly predicted proteins, leading to 20 wrongly predicted pairs. On the contrary, the highest accuracy we can achieve is as high as 80% if all the wrongly predicted 20 proteins pair among themselves leading to 10 wrongly predicted pairs.

To get a better statistical overview of NYCE's efficiency, rather than relying on the simple accuracy we compared it with random location prediction.

In a random location assignment test we assign one of the four locations with equal probability to each protein. We performed the random test 100000 times and computed the accuracy from the total number of correctly predicted pairs each time. The distribution of accuracy scores is shown in figure 7.1. We computed how many times the accuracy of the random test was equal to or greater than the NYCE accuracy score. By definition this is the p-value. Out of 100000 tests,

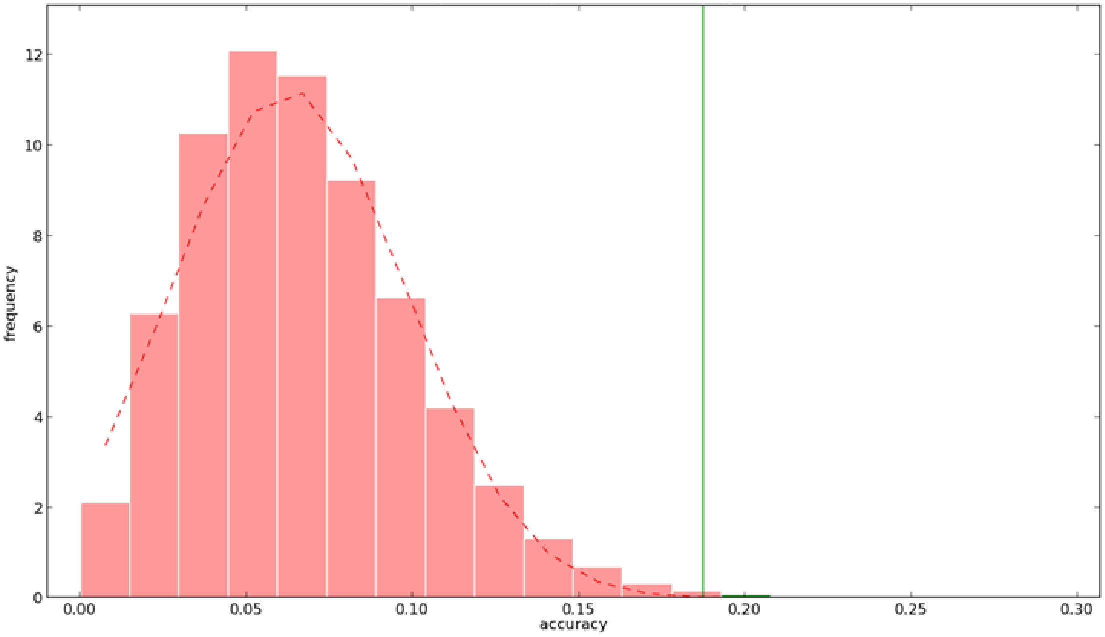


FIGURE 7.1: Comparison of NYCE with random assignment of location. Assignment of location to pairs of paralogs using NYCE is significantly better than random assignment. The green line represents the accuracy of NYCE versus the distribution of accuracies obtained from random simulations. In only 1499 cases out of 10^6 the result of the random test was better than NYCE.

the score was equal to or greater than the NYCE accuracy score for 4000 times. This results in a low p-value of $4000/100000 = 0.0015$, which indicates that NYCE performs significantly better compared to random location assignment.

7.4 Comparison of NYCE with other tools

To evaluate the performance of NYCE, we furthermore compared it with four other state-of-the-art subcellular location prediction tools: Yloc [144], Hum-mPLoc [215, 216], SherLoc [142] and PSORT-II [134]. For this purpose, these tools were selected based on the following criteria: public (web server) availability, reasonable response time, the ability to predict all three (nuclear, cytoplasmic and extracellular) location classes and the capacity to perform multi-class classification. None of the published methods for subcellular location prediction considers nucleocytoplasmic location as a separate class [217] and thus we will consider only the methods that can classify a protein into more than one class (nuclear and cytoplasmic) at the same time. Yloc and Hum-mPLoc have the capacity to classify proteins into multiple locations. Thus, proteins classified as nuclear and cytoplasmic by these tools are equivalent to the nucleocytoplasmic class of NYCE. Although the tools SherLoc and PSORT-II do not consider nucleocytoplasmic as a separate class, they provide a score for each class. We utilized the nuclear and cytoplasmic class score from these tools to generate a nucleocytoplasmic class association. For this purpose we applied a simple strategy that if the normalised nuclear + cytoplasmic score together is larger than 50% the protein is considered to be predicted as nucleocytoplasmic. It is important to mention that taking a cutoff value of more or less than 50% resulted in a poor performance of these tools regarding the total accuracy on proteins. Thus 50% is the optimized value for SherLoc and PSORT-II in the case of four-class classification. The result of this analysis is summarized in table 7.1.

Tool	Number of correctly predicted proteins	Accuracy on proteins (in %)	Number of correctly predicted pairs	Accuracy on pairs (in %)
NYCE	49	52.68	13	20.31
Yloc	42	45.16	3	4.68
Hum-mPLoc	35	37.63	3	4.68
SherLoc	40	43.01	0	0.00
PSORT II	37	39.78	10	15.62

Table 7.1: Performance of NYCE in comparison to other location prediction methods.

7.5 Discussion

In terms of total number of correctly predicted proteins, NYCE performs best. The tools Yloc and PSORT II are also reasonably close to the performance score of NYCE. For the paralog pairs, NYCE outperforms all the other tools. Performance of Yloc and SherLoc is reasonable in case of total accuracy but is bad for paralog protein pairs. This is not surprising given the fact that both Yloc and SherLoc use homology based methods for location analysis. On the other hand PSORT II does not depend on sequence homology for location detection, rather it uses sequence driven features and a k-nearest neighbors (kNN) classifier. This clearly indicates that homology is not necessarily the best criterion for location prediction.

Out of a total of 64 paralog pairs NYCE predicted the same location for both proteins in only 27 cases. This indicates that the NYCE method does not have a dependency on sequence homology. Furthermore, this analysis also showed that NYCE can discern proteins in different locations even when they show high similarity.

The methods used in this comparison are representative for other methods in terms of the underlying principle for location prediction. For example Yloc uses a combination of methods such as sequence homology, amino acid composition, PROSITE motifs, signal peptides, GO terms, etc. for location prediction. Similarly, HummPloc which is part of the Cell-PLoc [215] package, is specific to human proteins and applies a hybrid approach for prediction. The tool SherLoc integrates text-based features with several sequence-based classifiers originated from the MultiLoc [218, 219] prediction system. In general, the four location prediction methods used for this comparison cover other widely used methods for subcellular location prediction. Thus we can safely say that other existing prediction methods will not have a drastically higher performance.

Chapter 8

Discussion

After transcription and translation the resulting proteins are transported into proper subcellular locations to fulfill their functional purposes. Similar to the protein structure information, the subcellular location information of a protein should be encoded in its amino acid sequence and three-dimensional structure. Consequently, these signals are recognized by other sorting proteins and receptors. The prediction of subcellular location based on the identification of some of these signals (e.g. N-terminal signal peptides for extracellular, mitochondrial and chloroplast location) works quite well. However, our knowledge of these signals is incomplete. Furthermore, there are methods that infer the location of proteins based on the location of homologous proteins of experimentally verified location under the assumption that homologous proteins work in the same cellular locations. Such homology based methods lack accuracy and are not applicable in many cases, especially for novel proteins. In this thesis, we addressed the problem of protein subcellular location prediction from a different perspective and presented a novel method for subcellular location prediction based on protein residue exposure.

8.1 Residue exposure and subcellular location

We hypothesized that proteins evolve to match their environment by mutating their residues towards specific amino acid types whose side chains have physicochemical properties that agree to the subcellular location where the protein performs its major function. Since cellular compartments have different physicochemical environments, we reasoned that we could study amino acid composition to infer location. Moreover, we supposed that this process will depend on the levels of exposure of the amino acids involved and should be different for residues buried inside the protein, probably involved in interactions holding the protein together, or placed outside in contact with the solvent. Therefore, first we analysed how the composition of protein residues at various levels of exposure changes with the subcellular location of the protein. We demonstrated that the distribution of amino acids at different levels of exposure has signal about the location of proteins. The exposed residues are in direct contact to the subcellular environment. Thus for the proper functioning and interaction with other macromolecular entities such as DNA, RNA, etc. the exposed residues have to adapt according to the corresponding subcellular physicochemical properties [131]. Surprisingly, our analysis indicates that not only the exposed residues, but the buried residues also have location dependent roles. The buried residues, though not in direct contact to the subcellular environment, play an important role in protein stability. For example, compared to intracellular proteins, the extracellular proteins have to face an unspecified functional environment. To increase the stability, extracellular proteins require a more stable core [220, 221]. While location signals that guide protein sorting mechanisms are possibly the best predictor of a proteins' location, the residue exposure properties can be a useful predictor of subcellular location if such sorting signals are absent or unknown.

8.2 Two-step classification approach

To classify proteins into subcellular locations based on their residue exposure properties, we devised a two-step classification approach. We illustrated how a multi-class problem can be approached by using a two-step approach where first SVMs of different types score class membership for each of multiple classes and in a second step an ANN integrates the data and reassigns membership considering all scores from the SVMs.

Applying machine learning approaches in biological problems is a challenging task. In most cases the available datasets are unbalanced and/or small in size. For instance, the homology-reduced Höglund dataset [218], which is used for training and testing in several subcellular location prediction tools, contains 1411 cytoplasmic proteins while only 63 proteins are annotated as vacuolar. Applying a classification strategy (e.g. an ANN) on such unbalanced datasets with a high number of features can easily lead to a highly biased model. Our approach of creating a balanced dataset and using a two-step classification can be useful in such cases. In the proposed approach, the first step of classification (SVM) acts as a data compression technique. On such compressed data, for the final decision we can apply machine learning methods such as ANNs or a Bayesian approach.

8.3 Sequence homology is not location homology

For analysis and comparison, we collected pairs of homologous human proteins with different experimentally known locations. We found 64 paralog protein pairs that have different subcellular locations despite having sequence homology. It is important to note that we found this substantial amount of proteins although we considered only proteins with experimentally verified location annotation in one species (i.e. human) and exclusively four locations. We expect that considering many more subcellular locations and looking at other species will likely increase this number significantly. For example, by comparing subcellular location data of two yeast species *S. cerevisiae* and *S. pombe*, Yoshida et al. [59] identified several homologous proteins that have different subcellular location patterns. Similarly, Imai et al. [122] identified eighty protein pairs with significant homology but distinct subcellular locations in different animal species. Thus, there is growing evidence that "homologous sequences have similar locations" is not a canonic rule.

The method presented in this thesis can evaluate proteins independently of their homology. Thus it is a useful subcellular location predictor, especially in cases where homology is absent or homology is the only available criterion for location prediction.

8.4 Outlook

The focus of this work is on eukaryotic proteins from four subcellular locations. It is certainly possible to extend our classification system to other eukaryotic locations or for prokaryotic proteins. A major challenge in expanding the approach will be the lack of experimentally annotated location information and availability of three-dimensional protein structures. With the goal to analyse the whole human proteome, the Human Protein Atlas project is expect to be finished in 2015 [222]. It will provide high quality subcellular location information for proteins in 44 different normal human tissues and 20 different cancer types, as well as 46 different human cell lines. Similarly, a proteome level protein folding effort [223] called Human Proteome Folding Project, is in its second phase. We expect that such proteome level projects will provide reliable data for expansion of our approach. Using the optimized 'NYCE' algorithm it is possible to find protein families that have members with different location despite having high sequence similarity. Analysis of such protein families could reveal signals for yet unknown protein transport systems. Such information could be integrated with the protein-protein interaction (PPI) network to reveal the common partners of the proteins bearing a putative transporting signal, which then could be proposed as putative transport proteins. This will not only work as validation of PPI data but might also lead to the discovery of novel protein transport systems.

In the current era of high throughput systems biology, the significance of protein subcellular location information is becoming increasingly apparent. The direct correlation between protein mislocalization and diseases is well established [8, 224]. The proper function of a protein is subject to temporal and spatial conditions. Aberrant localization of proteins can disturb its normal function which can affect the cellular and metabolic pathway and lead to disease and cell death. Similarly, Xu et al. [225] have identified potential cancer biomarker proteins by comparing the subcellular location of proteins in normal and cancer tissues. Subcellular localization is also known to play a role in drug resistance in cancer cells [226]. Thus, the protein subcellular location information is becoming increasingly crucial not only for protein characterization but also to understand cellular mechanisms and disease. We expect that the novel approach presented in this thesis will eventually contribute to this goal.

Abstract

The proteins perform their functions in associated cellular locations. Therefore, subcellular location is a key-feature in the functional characterization of proteins. The experimental methods of determining a protein's subcellular location are costly, time consuming, error prone and can not cope with exponentially growing genomic and proteomic data. Therefore, computational prediction of protein subcellular location is a major effort in bioinformatics research. Subcellular location of a protein can be predicted either from its sequence by identifying the targeting peptide and motifs, or by homology to proteins of known location. Another approach, which is complementary, exploits the differences in amino acid composition of proteins associated to different cellular locations. This is an especially useful approach if motif and homology information are missing. In this study, we expand this approach taking into account amino acid composition at different levels of amino acid exposure.

Through careful selection and data integration we created a high quality dataset of proteins with known structure and location. The members of three subcellular location categories were considered: nuclear, cytoplasmic and extracellular, plus the extra category nucleocytoplasmic, accounting for the fact that a large number of proteins shuttle between nucleus and cytoplasm. We explored the relationship between residue exposure and protein subcellular location. The analysis demonstrated that amino acids at different levels of exposure have signal about the location of proteins.

For the classification purpose we applied a novel approach of two stage classification. At stage one, multiple Support Vector Machines (SVMs) were trained to score eukaryotic protein sequences for membership to each location class. In stage two, an artificial neural network (ANN) was used to propose a category from the scores assigned to the four locations in stage one. The method reaches an accuracy of 68% when using as input 3D-derived values of amino acid exposure. Calibration of the method using predicted values of amino acid exposure allows classifying proteins without 3D-information with an accuracy of 62%. The algorithm is implemented as the web server 'NYCE'.

We compared the performance of NYCE against other state-of-the-art subcellular location prediction tools. The comparison revealed the fact that 'NYCE' performs reasonably well compared to other tools, though using a limited set of information. A major challenge of protein subcellular location prediction methods based on homology is that there are very similar proteins that act in different subcellular locations. Using pairs of paralog proteins experimentally known to be in different locations, we demonstrated that our algorithm can evaluate proteins independently of their homology. NYCE can discern proteins in different locations even if they share high levels of identity whereas other tools fail to do so.

Zusammenfassung

Proteine können ihre Funktion nur in bestimmten intrazellulären Kompartimenten erfüllen, deshalb ist die subzelluläre Lokalisation ein wichtiges Hauptmerkmal in der funktionellen Charakterisierung von Proteinen. Die experimentellen Methoden zur Bestimmung der subzellulären Lokalisation von Proteinen sind teuer, zeitintensiv, fehleranfällig und können nicht mit der exponentiell anwachsenden Menge an genomischen und proteomischen Daten mithalten. Aus diesem Grund ist die computergestützte Vorhersage der intrazellulären Lokalisation von Proteinen ein wichtiges Ziel der bioinformatischen Forschung. Die Lokalisation eines Proteins kann entweder aus dessen Sequenz durch die Analyse von Zielsequenzen und -motiven vorhergesagt werden oder durch das Heranziehen homologer Proteine deren Lokalisation schon bekannt ist. Ein anderer, komplementärer Ansatz nutzt die Aminosäurezusammensetzung von verschieden lokalisierten Proteinen. In dieser Arbeit erweitern wir diesen Ansatz, indem wir die Aminnosäurezusammensetzung in Zusammenhang damit betrachten, wie gut die Aminosäuren aufgrund der Proteinstruktur von außen zugänglich sind.

Es wurden drei Kategorien der subzellulären Lokalisation in die Untersuchungen einbezogen: nukleär, zytoplasmatisch und extrazellulär. Zusätzlich wurde die Kategorie nukleo-zytoplasmatisch eingeführt, welche Proteine enthält, die sich zwischen Zytoplasma und dem Nukleus bewegen. Wir haben einen qualitativ hochwertigen Datensatz zusammengestellt, der Proteine mit bekannter Struktur und Lokalisation enthält und den Zusammenhang zwischen der Zugänglichkeit der Aminosäuren und der subzellulären Lokalisation des Proteins untersucht. Diese Analyse hat gezeigt, dass Aminosäuren mit verschiedenen Zugänglichkeiten zur Vorhersage der Lokalisation von Proteinen genutzt werden können.

Zum Zweck der Klassifizierung haben wir einen neuartigen Ansatz, basierend auf einer zweistufigen Klassifizierung, verwendet. In der ersten Stufe werden Support Vector Machines (SVMs) trainiert, die Wahrscheinlichkeit der Zugehörigkeit (Score) für alle Klassen anhand der Proteinsequenzen zu berechnen. Die zweite Stufe, ein künstliches neuronales Netzwerk (KNN), wird benutzt um eine Kategorie auf der Grundlage der vorher berechneten Scores für die vier möglichen Lokalisationen vorzuschlagen. Diese Methode erreicht eine Präzision von 68% wenn auf 3D-Strukturen basierende Werte für die Zugänglichkeit der Aminosäuren benutzt werden. Die Kalibrierung der Methode mithilfe theoretisch berechneter Werte für die Zugänglichkeit der Aminosäuren ermöglicht eine Klassifizierung der Proteine ohne 3D-Information mit einer Präzision von 62%. Der Algorithmus wurde als der Webserver “NYCE” implementiert.

Ein Vergleich von “NYCE” mit anderen modernen Vorhersageprogrammen zeigte, dass obwohl “NYCE” nur die Proteinsequenz und somit ein sehr beschränktes Set an Informationen nutzt, die Leistung vergleichsweise gut ist. Ein großes Problem der auf Homologie basierenden

Vorhersageprogramme ist die Existenz von Proteinen mit sehr ähnlicher Sequenz aber unterschiedlicher subzellulärer Lokalisation. Anhand paraloger Proteine, welche unterschiedliche Lokalisation aufweisen, konnten wir zeigen dass “NYCE” - im Gegensatz zu anderen Vorhersageprogrammen - zwischen Proteinen mit großer Sequenzähnlichkeit aber verschiedener Lokalisation unterscheiden kann.

Unser Ansatz kann in Zukunft für die Vorhersage der Lokalisation von Proteinen in anderen Kompartimenten und in nicht-eukaryotischen Organismen nützlich sein. Dennoch ist eine gewissenhafte Auswahl an Trainingsdaten notwendig, um verlässliche Resultate in der Vorhrsage zu erzielen. Wir erwarten, dass solch eine Erweiterung unserer Methode durch die wachsende Anzahl von in Datenbanken verfügbaren Proteinstrukturen und Proteinen mit experimentell bestätigter Lokalisation erleichtert wird.

Bibliography

[1] L Giot, J S Bader, C Brouwer, A Chaudhuri, B Kuang, Y Li, Y L Hao, C E Ooi, B Godwin, E Vitols, G Vijayadamodar, P Pochart, H Machineni, M Welsh, Y Kong, B Zerhusen, R Malcolm, Z Varrone, A Collis, M Minto, S Burgess, L McDaniel, E Stimpson, F Spriggs, J Williams, K Neurath, N Ioime, M Agee, E Voss, K Furtak, R Renzulli, N Aanensen, S Carrolla, E Bickelhaupt, Y Lazovatsky, A DaSilva, J Zhong, C a Stanyon, R L Finley, K P White, M Braverman, T Jarvie, S Gold, M Leach, J Knight, R a Shimkets, M P McKenna, J Chant, and J M Rothberg. A protein interaction map of Drosophila melanogaster. *Science (New York, N.Y.)*, 302(5651): 1727–36, December 2003.

[2] Chang Jin Shin, Simon Wong, Melissa J Davis, and Mark A Ragan. Protein-protein interaction as a predictor of subcellular location. *BMC systems biology*, 3:28, 2009.

[3] F Matsuzaki. Asymmetric division of Drosophila neural stem cells: a basis for neural diversity. *Current opinion in neurobiology*, 10(1):38–44, March 2000.

[4] Juergen a Knoblich. Mechanisms of asymmetric stem cell division. *Cell*, 132 (4):583–97, February 2008.

[5] D E Nelson, A E C Ihekwaba, M Elliott, J R Johnson, C A Gibney, B E Foreman, G Nelson, V See, C A Horton, D G Spiller, S W Edwards, H P McDowell, J F Unitt, E Sullivan, R Grimley, N Benson, D Broomhead, D B Kell, and M R H White. Oscillations in NF-kappaB signaling control the

dynamics of gene expression. *Science (New York, N.Y.)*, 306(5696):704–8, October 2004.

[6] Candida Vaz, Arvind Singh Mer, Alok Bhattacharya, and Ramakrishna Ramaswamy. MicroRNAs modulate the dynamics of the NF-κB signaling pathway. *PloS one*, 6(11):e27774, January 2011.

[7] Mien-Chie Hung and Wolfgang Link. Protein localization in disease and therapy. *Journal of cell science*, 124(Pt 20):3381–92, October 2011.

[8] Solip Park, Jae-Seong Yang, Young-Eun Shin, Juyong Park, Sung Key Jang, and Sanguk Kim. Protein localization as a principal feature of the etiology and comorbidity of genetic diseases. *Molecular systems biology*, 7(494):494, May 2011.

[9] Snezana Djordjevic, Xiaoxuan Zhang, Mark Bartlam, Sheng Ye, Zihe Rao, and Christopher J Danpure. Structural implications of a G170R mutation of alanine:glyoxylate aminotransferase that is associated with peroxisome-to-mitochondrion mistargeting. *Acta crystallographica. Section F, Structural biology and crystallization communications*, 66(Pt 3):233–6, March 2010.

[10] S W Edwards, C M Tan, and L E Limbird. Localization of G-protein-coupled receptors in health and disease. *Trends in pharmacological sciences*, 21(8): 304–8, August 2000.

[11] Joris H Robben, Nine V A M Knoers, and Peter M T Deen. Cell biological aspects of the vasopressin type-2 receptor and aquaporin 2 water channel in nephrogenic diabetes insipidus. *American journal of physiology. Renal physiology*, 291(2):F257–70, August 2006.

[12] Hugo F Mendes, Jacqueline van der Spuy, J Paul Chapple, and Michael E Cheetham. Mechanisms of cell death in rhodopsin retinitis pigmentosa: implications for therapy. *Trends in molecular medicine*, 11(4):177–85, April 2005.

[13] P Michael Conn, Alfredo Ulloa-Aguirre, Joel Ito, and Jo Ann Janovick. G protein-coupled receptor trafficking in health and disease: lessons learned to prepare for therapeutic mutant rescue in vivo. *Pharmacological reviews*, 59 (3):225–50, September 2007.

[14] Catherine M Cowan and Lynn A Raymond. Selective neuronal degeneration in Huntington's disease. *Current topics in developmental biology*, 75:25–71, January 2006.

[15] Uwe Ueberham, Elke Ueberham, Hildegard Gruschka, and Thomas Arendt. Altered subcellular location of phosphorylated Smads in Alzheimer's disease. *The European journal of neuroscience*, 24(8):2327–34, October 2006.

[16] Brian R Hoover, Miranda N Reed, Jianjun Su, Rachel D Penrod, Linda A Kotilinek, Marianne K Grant, Rose Pitstick, George A Carlson, Lorene M Lanier, Li-Lian Yuan, Karen H Ashe, and Dezhi Liao. Tau mislocalization to dendritic spines mediates synaptic dysfunction independently of neurodegeneration. *Neuron*, 68(6):1067–81, December 2010.

[17] Vivek P Patel and Charleen T Chu. Nuclear transport, oxidative stress, and neurodegeneration. *International journal of clinical and experimental pathology*, 4(3):215–29, March 2011.

[18] Charleen T Chu, Edward D Plowey, Ying Wang, Vivek Patel, and Kelly L Jordan-Sciutto. Location, location, location: altered transcription factor trafficking in neurodegeneration. *Journal of neuropathology and experimental neurology*, 66(10):873–83, October 2007.

[19] S H Liang and M F Clarke. Regulation of p53 localization. *European journal of biochemistry / FEBS*, 268(10):2779–83, May 2001.

[20] Tweeny R Kau, Jeffrey C Way, and Pamela A Silver. Nuclear transport and cancer: from mechanism to intervention. *Nature reviews. Cancer*, 4(2): 106–17, February 2004.

[21] Yohannes Mebratu and Yohannes Tesfaigzi. How ERK1/2 activation controls cell proliferation and cell death: Is subcellular localization the answer? *Cell cycle (Georgetown, Tex.)*, 8(8):1168–75, April 2009.

[22] James J Manfredi. An identity crisis for a cancer gene: subcellular location determines ASPP1 function. *Cancer cell*, 18(5):409–10, November 2010.

[23] Céline Charlot, Hélène Dubois-Pot, Tsvetan Serchov, Yves Tourrette, and Bohdan Wasylyk. A review of post-translational modifications and subcellular localization of Ets transcription factors: possible connection with cancer and involvement in the hypoxic response. *Methods in molecular biology (Clifton, N.J.)*, 647:3–30, January 2010.

[24] Hyung Seok Park, Ji Min Park, Seho Park, Junghoon Cho, Seung Il Kim, and Byeong-Woo Park. Subcellular localization of Mdm2 expression and prognosis of breast cancer. *International journal of clinical oncology*, November 2013.

[25] Piotr Mamczur, Andrzej Gamian, Jerzy Kolodziej, Piotr Dziegiel, and Dariusz Rakus. Nuclear localization of aldolase A correlates with cell proliferation. *Biochimica et biophysica acta*, 1833(12):2812–22, December 2013.

[26] Z Y Yang, N D Perkins, T Ohno, E G Nabel, and G J Nabel. The p21 cyclin-dependent kinase inhibitor suppresses tumorigenicity in vivo. *Nature medicine*, 1(10):1052–6, October 1995.

[27] Giuseppe Viglietto, Maria Letizia Motti, Paola Bruni, Rosa Marina Melillo, Amelia D'Alessio, Daniela Califano, Floriana Vinci, Gennaro Chiappetta, Philip Tsichlis, Alfonso Bellacosa, Alfredo Fusco, and Massimo Santoro. Cytoplasmic relocalization and inhibition of the cyclin-dependent kinase inhibitor p27(Kip1) by PKB/Akt-mediated phosphorylation in breast cancer. *Nature medicine*, 8(10):1136–44, October 2002.

[28] Shuji Ogino, Kaori Shima, Katsuhiko Nosho, Natsumi Irahara, Yoshifumi Baba, Brian M Wolpin, Edward L Giovannucci, Jeffrey A Meyerhardt, and

Charles S Fuchs. A cohort study of p27 localization in colon cancer, body mass index, and patient survival. *Cancer epidemiology, biomarkers & prevention : a publication of the American Association for Cancer Research, cosponsored by the American Society of Preventive Oncology*, 18(6):1849–58, June 2009.

[29] Jennifer L Gardy and Fiona S L Brinkman. Methods for predicting bacterial protein subcellular localization. *Nature reviews. Microbiology*, 4(10):741–51, October 2006.

[30] P Bork, T Dandekar, Y Diaz-Lazcoz, F Eisenhaber, M Huynen, and Y Yuan. Predicting function: from genes to genomes and back. *Journal of molecular biology*, 283:707–725, 1998.

[31] Rita Casadio, Pier Luigi Martelli, and Andrea Pierleoni. The prediction of protein subcellular localization from sequence: a shortcut to functional genome annotation. *Briefings in functional genomics & proteomics*, 7(1): 63–73, January 2008.

[32] Marcia Gamberini, Ricardo M Gómez, Marina V Atzingen, Elizabeth A L Martins, Silvio A Vasconcellos, Eliete C Romero, Luciana C C Leite, Paulo L Ho, and Ana L T O Nascimento. Whole-genome analysis of Leptospira interrogans to identify potential vaccine candidates against leptospirosis. *FEMS microbiology letters*, 244(2):305–13, March 2005.

[33] Nestor Solis and Stuart J Cordwell. Current methodologies for proteomics of bacterial surface-exposed and cell envelope proteins. *Proteomics*, 11(15): 3169–89, August 2011.

[34] Emmanuel G Reynaud, Miguel A Andrade, Fabien Bonneau, Thi Bach Nga Ly, Michael Knop, Klaus Scheffzek, and Rainer Pepperkok. Human Lsg1 defines a family of essential GTPases that correlates with the evolution of compartmentalization. *BMC Biology*, 3:21, 2005.

[35] Stuart J Cordwell. Technologies for bacterial surface proteomics. *Current opinion in microbiology*, 9(3):320–9, June 2006.

[36] Guido Grandi. Bacterial surface proteins and vaccines. *F1000 biology reports*, 2, 2010.

[37] Mathias Uhlén, Erik Björling, Charlotta Agaton, Cristina Al-Khalili Szigyarto, Bahram Amini, Elisabet Andersen, Ann-Catrin Andersson, Pia Angelidou, Anna Asplund, Caroline Asplund, Lisa Berglund, Kristina Bergström, Harry Brumer, Dijana Cerjan, Marica Ekström, Adila Elobeid, Cecilia Eriksson, Linn Fagerberg, Ronny Falk, Jenny Fall, Mattias Forsberg, Marcus Gry Björklund, Kristoffer Gumbel, Asif Halimi, Inga Hallin, Carl Hamsten, Marianne Hansson, My Hedhammar, Görel Hercules, Caroline Kampf, Karin Larsson, Mats Lindskog, Wald Lodewyckx, Jan Lund, Joakim Lundeberg, Kristina Magnusson, Erik Malm, Peter Nilsson, Jenny Odling, Per Oksvold, Ingmarie Olsson, Emma Oster, Jenny Ottosson, Linda Paavilainen, Anja Persson, Rebecca Rimini, Johan Rockberg, Marcus Runeson, Asa Sivertsson, Anna Sköllermo, Johanna Steen, Maria Stenvall, Fredrik Sterky, Sara Strömberg, Må rten Sundberg, Hanna Tegel, Samuel Tourle, Eva Wahlund, Annelie Waldén, Jinghong Wan, Henrik Wernérus, Joakim Westberg, Kenneth Wester, Ulla Wrethagen, Lan Lan Xu, Sophia Hober, and Fredrik Pontén. A human protein atlas for normal and cancer tissues based on antibody proteomics. *Molecular & cellular proteomics : MCP*, 4: 1920–1932, 2005.

[38] F Pontén, K Jirström, and M Uhlen. The Human Protein Atlas–a tool for pathology. *The Journal of pathology*, 216(4):387–93, December 2008.

[39] Mathias Uhlen, Per Oksvold, Linn Fagerberg, Emma Lundberg, Kalle Jonasson, Mattias Forsberg, Martin Zwahlen, Caroline Kampf, Kenneth Wester, Sophia Hober, Henrik Wernerus, Lisa Björling, and Fredrik Ponten. Towards a knowledge-based Human Protein Atlas. *Nature biotechnology*, 28 (12):1248–50, December 2010.

[40] Laurent Barbe, Emma Lundberg, Per Oksvold, Anna Stenius, Erland Lewin, Erik Björling, Anna Asplund, Fredrik Pontén, Hjalmar Brismar, Mathias Uhlén, and Helene Andersson-Svahn. Toward a confocal subcellular atlas of

the human proteome. *Molecular & cellular proteomics : MCP*, 7:499–508, 2008.

[41] Mathias Uhlen and Fredrik Ponten. Antibody-based proteomics for human tissue profiling. *Molecular & cellular proteomics : MCP*, 4(4):384–93, April 2005.

[42] J A Heyman, J Cornthwaite, L Foncerrada, J R Gilmore, E Gontang, K J Hartman, C L Hernandez, R Hood, H M Hull, W Y Lee, R Marcil, E J Marsh, K M Mudd, M J Patino, T J Purcell, J J Rowland, M L Sindici, and J P Hoeffler. Genome-scale cloning and expression of individual open reading frames using topoisomerase I-mediated ligation. *Genome research*, 9(4):383–92, April 1999.

[43] M S Longtine, A McKenzie, D J Demarini, N G Shah, A Wach, A Brachat, P Philippsen, and J R Pringle. Additional modules for versatile and economical PCR-based gene deletion and modification in Saccharomyces cerevisiae. *Yeast (Chichester, England)*, 14(10):953–61, July 1998.

[44] H G Sutherland, G K Mumford, K Newton, L V Ford, R Farrall, G Dellaire, J F Cáceres, and W a Bickmore. Large-scale identification of mammalian proteins localized to nuclear sub-compartments. *Human molecular genetics*, 10(18):1995–2011, September 2001.

[45] Anuj Kumar, Seema Agarwal, John A Heyman, Sandra Matson, Matthew Heidtman, Stacy Piccirillo, Lara Umansky, Amar Drawid, Ronald Jansen, Yang Liu, Kei-Hoi Cheung, Perry Miller, Mark Gerstein, G Shirleen Roeder, and Michael Snyder. Subcellular localization of the yeast proteome. *Genes & development*, 16(6):707–19, March 2002.

[46] Michael Sauer, Tomasz Paciorek, Eva Benková, and Jirí Friml. Immunocytochemical techniques for whole-mount in situ protein localization in plants. *Nature protocols*, 1(1):98–103, January 2006.

[47] P Ross-Macdonald, P S Coelho, T Roemer, S Agarwal, A Kumar, R Jansen, K H Cheung, A Sheehan, D Symoniatis, L Umansky, M Heidtman, F K Nelson, H Iwasaki, K Hager, M Gerstein, P Miller, G S Roeder, and M Snyder. Large-scale analysis of the yeast genome by transposon tagging and gene disruption. *Nature*, 402(6760):413–8, November 1999.

[48] Won-Ki Huh, James V Falvo, Luke C Gerke, Adam S Carroll, Russell W Howson, Jonathan S Weissman, and Erin K O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425(6959):686–91, October 2003.

[49] M Ormö, A B Cubitt, K Kallio, L A Gross, R Y Tsien, and S J Remington. Crystal structure of the Aequorea victoria green fluorescent protein. *Science (New York, N.Y.)*, 273(5280):1392–5, September 1996.

[50] M Chalfie, Y Tu, G Euskirchen, W W Ward, and D C Prasher. Green fluorescent protein as a marker for gene expression. *Science (New York, N.Y.)*, 263(5148):802–5, February 1994.

[51] R K Niedenthal, L Riles, M Johnston, and J H Hegemann. Green fluorescent protein as a marker for gene expression and subcellular localization in budding yeast. *Yeast (Chichester, England)*, 12(8):773–86, June 1996.

[52] D Q Ding, Yuki Tomita, Ayumu Yamamoto, Yuji Chikashige, Tokuko Haraguchi, and Yasushi Hiraoka. Large-scale screening of intracellular protein localization in living fission yeast cells by the use of a GFP-fusion genomic DNA library. *Genes to cells : devoted to molecular & cellular mechanisms*, 5(3):169–90, March 2000.

[53] J C Simpson, R Wellenreuther, A Poustka, R Pepperkok, and S Wiemann. Systematic subcellular localization of novel proteins identified by large-scale cDNA sequencing. *EMBO reports*, 1(3):287–92, September 2000.

[54] V Wood, R Gwilliam, M-A Rajandream, M Lyne, R Lyne, A Stewart, J Sgouros, N Peat, J Hayles, S Baker, D Basham, S Bowman, K Brooks,

D Brown, S Brown, T Chillingworth, C Churcher, M Collins, R Connor, A Cronin, P Davis, T Feltwell, A Fraser, S Gentles, A Goble, N Hamlin, D Harris, J Hidalgo, G Hodgson, S Holroyd, T Hornsby, S Howarth, E J Huckle, S Hunt, K Jagels, K James, L Jones, M Jones, S Leather, S McDonald, J McLean, P Mooney, S Moule, K Mungall, L Murphy, D Niblett, C Odell, K Oliver, S O'Neil, D Pearson, M A Quail, E Rabbinowitsch, K Rutherford, S Rutter, D Saunders, K Seeger, S Sharp, J Skelton, M Simmonds, R Squares, S Squares, K Stevens, K Taylor, R G Taylor, A Tivey, S Walsh, T Warren, S Whitehead, J Woodward, G Volckaert, R Aert, J Robben, B Grymonprez, I Weltjens, E Vanstreels, M Rieger, M Schäfer, S Müller-Auer, C Gabel, M Fuchs, A Düsterhöft, C Fritzc, E Holzer, D Moestl, H Hilbert, K Borzym, I Langer, A Beck, H Lehrach, R Reinhardt, T M Pohl, P Eger, W Zimmermann, H Wedler, R Wambutt, B Purnelle, A Goffeau, E Cadieu, S Dréano, S Gloux, V Lelaure, S Mottier, F Galibert, S J Aves, Z Xiang, C Hunt, K Moore, S M Hurst, M Lucas, M Rochet, C Gaillardin, V A Tallada, A Garzon, G Thode, R R Daga, L Cruzado, J Jimenez, M Sánchez, F del Rey, J Benito, A Domínguez, J L Revuelta, S Moreno, J Armstrong, S L Forsburg, L Cerutti, T Lowe, W R McCombie, I Paulsen, J Potashkin, G V Shpakovski, D Ussery, B G Barrell, P Nurse, and L Cerrutti. The genome sequence of Schizosaccharomyces pombe. *Nature*, 415(6874):871–80, February 2002.

[55] Sina Ghaemmaghami, Won-Ki Huh, Kiowa Bower, Russell W Howson, Archana Belle, Noah Dephoure, Erin K O'Shea, and Jonathan S Weissman. Global analysis of protein expression in yeast. *Nature*, 425(6959):737–41, October 2003.

[56] J Michael Cherry, Eurie L Hong, Craig Amundsen, Rama Balakrishnan, Gail Binkley, Esther T Chan, Karen R Christie, Maria C Costanzo, Selina S Dwight, Stacia R Engel, Dianna G Fisk, Jodi E Hirschman, Benjamin C Hitz, Kalpana Karra, Cynthia J Krieger, Stuart R Miyasato, Rob S Nash, Julie Park, Marek S Skrzypek, Matt Simison, Shuai Weng, and Edith D

Wong. Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic acids research*, 40(Database issue):D700–5, January 2012.

[57] M P Rout, J D Aitchison, A Suprapto, K Hjertaas, Y Zhao, and B T Chait. The yeast nuclear pore complex: composition, architecture, and transport mechanism. *The Journal of cell biology*, 148(4):635–51, February 2000.

[58] P A Wigge, O N Jensen, S Holmes, S Souès, M Mann, and J V Kilmartin. Analysis of the Saccharomyces spindle pole by matrix-assisted laser desorption/ionization (MALDI) mass spectrometry. *The Journal of cell biology*, 141(4):967–77, May 1998.

[59] Akihisa Matsuyama, Ritsuko Arai, Yoko Yashiroda, Atsuko Shirai, Ayako Kamata, Shigeko Sekido, Yumiko Kobayashi, Atsushi Hashimoto, Makiko Hamamoto, Yasushi Hiraoka, Sueharu Horinouchi, and Minoru Yoshida. ORFeome cloning and global analysis of protein localization in the fission yeast Schizosaccharomyces pombe. *Nature biotechnology*, 24(7):841–7, July 2006.

[60] K S Ullman, M A Powers, and D J Forbes. Nuclear export receptors: from importin to exportin. *Cell*, 90(6):967–70, September 1997.

[61] Minoru Yoshida and Shelley Sazer. Nucleocytoplasmic transport and nuclear envelope integrity in the fission yeast Schizosaccharomyces pombe. *Methods (San Diego, Calif.)*, 33(3):226–38, July 2004.

[62] M Fornerod, M Ohno, M Yoshida, and I W Mattaj. CRM1 is an export receptor for leucine-rich nuclear export signals. *Cell*, 90(6):1051–60, September 1997.

[63] Markus Islinger, Ka Wan Li, Jürgen Seitz, Alfred Völkl, and Georg H Lüers. Hitchhiking of Cu/Zn superoxide dismutase to peroxisomes–evidence for a natural piggyback import mechanism in mammals. *Traffic (Copenhagen, Denmark)*, 10(11):1711–21, November 2009.

[64] Gilad Twig, Solomon A Graf, Jakob D Wikstrom, Hibo Mohamed, Sarah E Haigh, Alvaro Elorza, Motti Deutsch, Naomi Zurgil, Nicole Reynolds, and

Orian S Shirihai. Tagging and tracking individual networks within a complex mitochondrial web with photoactivatable GFP. *American journal of physiology. Cell physiology*, 291(1):C176–84, July 2006.

[65] Satu Passinen, Jan Valkila, Tommi Manninen, Heimo Syvälä, and Timo Ylikomi. The C-terminal half of Hsp90 is responsible for its cytoplasmic localization. *European journal of biochemistry / FEBS*, 268(20):5337–42, October 2001.

[66] C L Thomas and A J Maule. Limitations on the use of fused green fluorescent protein to investigate structure-function relationships for the cauliflower mosaic virus movement protein. *The Journal of general virology*, 81(Pt 7): 1851–5, July 2000.

[67] Y. Kuroda, N. Suzuki, and T. Kataoka. The effect of posttranslational modifications on the interaction of Ras2 with adenylyl cyclase. *Science (New York, N.Y.)*, 259(5095):683–6, January 1993.

[68] S Bhattacharya, L Chen, J R Broach, and S Powers. Ras membrane targeting is essential for glucose signaling but not for viability in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, 92(7): 2984–8, March 1995.

[69] S J Gould, G A Keller, M Schneider, S H Howell, L J Garrard, J M Goodman, B Distel, H Tabak, and S Subramani. Peroxisomal protein import is conserved between yeast, plants, insects and mammals. *The EMBO journal*, 9(1):85–90, January 1990.

[70] H R Pelham, K G Hardwick, and M J Lewis. Sorting of soluble ER proteins in yeast. *The EMBO journal*, 7(6):1757–62, June 1988.

[71] M A van Berkel, L H Caro, R C Montijn, and F M Klis. Glucosylation of chimeric proteins in the cell wall of Saccharomyces cerevisiae. *FEBS letters*, 349(1):135–8, July 1994.

[72] M Dreger, L Bengtsson, T Schöneberg, H Otto, and F Hucho. Nuclear envelope proteomics: novel integral membrane proteins of the inner nuclear membrane. *Proceedings of the National Academy of Sciences of the United States of America*, 98(21):11943–8, October 2001.

[73] Olof Emanuelsson. Predicting protein subcellular localisation from amino acid sequence information. *Briefings in bioinformatics*, 3(4):361–76, December 2002.

[74] G Schatz and B Dobberstein. Common principles of protein translocation across membranes. *Science*, 271:1519–1526, 1996.

[75] K Ito. The major pathways of protein translocation across membranes. *Genes to cells : devoted to molecular & cellular mechanisms*, 1(4):337–46, April 1996.

[76] Pamela A. Silver. How proteins enter the nucleus. *Cell*, 64(3):489–497, 1991.

[77] E A Nigg. Nucleocytoplasmic transport: signals, mechanisms and regulation. *Nature*, 386(6627):779–87, April 1997.

[78] Ruiwen Wang and Michael G. Brattain. The maximal size of protein to diffuse through the nuclear pore is larger than 60kDa. *FEBS letters*, 581 (17):3164–70, July 2007.

[79] Laura J Mauro and Jack E Dixon. 'Zip codes' direct intracellular protein tyrosine phosphatases to the correct cellular 'address'. *Trends in biochemical sciences*, 19(4):151–5, April 1994.

[80] R M Stroud and P Walter. Signal sequence recognition and protein targeting. *Current opinion in structural biology*, 9(6):754–9, December 1999.

[81] Nikolaj Zuleger, Alastair R W Kerr, and Eric C Schirmer. Many mechanisms, one entrance: membrane protein translocation into the nucleus. *Cellular and molecular life sciences : CMLS*, February 2012.

[82] M O Lively. Signal peptidases in protein biosynthesis and intracellular transport. *Current opinion in cell biology*, 1(6):1188–93, December 1989.

[83] Ross E. Dalbey and G Von Heijne. Signal peptidases in prokaryotes and eukaryotes–a new protease family. *Trends in biochemical sciences*, 17(11): 474–8, November 1992.

[84] G von Heijne, J Steppuhn, and R G Herrmann. Domain structure of mitochondrial and chloroplast targeting peptides. *European journal of biochemistry / FEBS*, 180:535–545, 1989.

[85] G von Heijne. The signal peptide. *The Journal of membrane biology*, 115 (3):195–201, May 1990.

[86] Y Fujiwara and M Asogawa. Prediction of subcellular localizations using amino acid composition and order. *Genome informatics. International Conference on Genome Informatics*, 12:103–12, January 2001.

[87] G von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic acids research*, 14(11):4683–90, June 1986.

[88] H Nielsen, J Engelbrecht, S Brunak, and G von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein engineering*, 10(1):1–6, January 1997.

[89] H Nielsen and A Krogh. Prediction of signal peptides and signal anchors by a hidden Markov model. *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology*, 6:122–30, January 1998.

[90] Nancy Y Yu, James R Wagner, Matthew R Laird, Gabor Melli, Sébastien Rey, Raymond Lo, Phuong Dao, S Cenk Sahinalp, Martin Ester, Leonard J Foster, and Fiona S L Brinkman. PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. *Bioinformatics*, 26:1608–1615, 2010.

[91] Thomas Nordahl Petersen, Sø ren Brunak, Gunnar von Heijne, and Henrik Nielsen. SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nature methods*, 8(10):785–6, January 2011.

[92] Sheila M Reynolds, Lukas Käll, Michael E Riffle, Jeff A Bilmes, and William Stafford Noble. Transmembrane topology and signal peptide prediction using dynamic bayesian networks. *PLoS computational biology*, 4 (11):e1000213, November 2008.

[93] Hå kan Viklund, Andreas Bernsel, Marcin Skwark, and Arne Elofsson. SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology. *Bioinformatics (Oxford, England)*, 24(24):2928–9, December 2008.

[94] Timothy Nugent and David T Jones. Transmembrane protein topology prediction using support vector machines. *BMC bioinformatics*, 10:159, January 2009.

[95] Olof Emanuelsson, Sø ren Brunak, Gunnar von Heijne, and Henrik Nielsen. Locating proteins in the cell using TargetP, SignalP and related tools. *Nature protocols*, 2(4):953–71, January 2007.

[96] O Emanuelsson, H Nielsen, S Brunak, and G von Heijne. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *Journal of molecular biology*, 300(4):1005–16, July 2000.

[97] Setsuro Matsuda, Jean-philippe Vert, Hiroto Saigo, Nobuhisa Ueda, Hiroyuki Toh, and Tatsuya Akutsu. A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein science : a publication of the Protein Society*, 14(11):2804–13, November 2005.

[98] Michelle S Scott, Peter V Troshin, and Geoffrey J Barton. NoD: a Nucleolar localization sequence detector for eukaryotic and viral proteins. *BMC bioinformatics*, 12(1):317, January 2011.

[99] Alex N Nguyen Ba, Anastassia Pogoutse, Nicholas Provart, and Alan M Moses. NLStradamus: a simple Hidden Markov Model for nuclear localization signal prediction. *BMC bioinformatics*, 10:202, January 2009.

[100] Manuel G. Claros and Pierre Vincens. Computational method to predict mitochondrially imported proteins and their targeting sequences. *European journal of biochemistry / FEBS*, 241(3):779–86, November 1996.

[101] Rajesh Nair, Phil Carter, and Burkhard Rost. NLSdb: database of nuclear localization signals. *Nucleic acids research*, 31(1):397–9, January 2003.

[102] Khar Heng Choo, Tin Wee Tan, and Shoba Ranganathan. SPdb–a signal peptide database. *BMC bioinformatics*, 6(1):249, January 2005.

[103] Dmitry N Ivankov, Samuel H Payne, Michael Y Galperin, Stefano Bonissone, Pavel A Pevzner, and Dmitrij Frishman. How many signal peptides are there in bacteria? *Environmental microbiology*, 15(4):983–90, April 2013.

[104] L J Zhao and R Padmanabhan. Nuclear transport of adenovirus DNA polymerase is facilitated by interaction with preterminal protein. *Cell*, 55(6): 1005–15, December 1988.

[105] K Melén and I Julkunen. Nuclear cotransport mechanism of cytoplasmic human MxB protein. *The Journal of biological chemistry*, 272(51):32353–9, December 1997.

[106] B N Kholodenko, J B Hoek, and H V Westerhoff. Why cytoplasmic signalling proteins should be recruited to cell membranes. *Trends in cell biology*, 10 (5):173–8, May 2000.

[107] Boris N Kholodenko. Four-dimensional organization of protein kinase signaling cascades: the roles of diffusion, endocytosis and molecular motors. *The Journal of experimental biology*, 206(Pt 12):2073–82, June 2003.

[108] Eric J Arnoys and John L Wang. Dual localization: proteins in extracellular and intracellular compartments. *Acta histochemica*, 109(2):89–110, January 2007.

[109] Kylie M Wagstaff and David a Jans. Importins and beyond: non-conventional nuclear transport mechanisms. *Traffic (Copenhagen, Denmark)*, 10(9):1188–98, September 2009.

[110] Frank Eisenhaber and Peer Bork. Wanted: subcellular localization of proteins based on sequence. *Trends in cell biology*, 8(4):169–70, April 1998.

[111] Rajesh Nair and Burkhard Rost. Sequence conserved for subcellular localization. *Protein science : a publication of the Protein Society*, 11(12):2836–47, December 2002.

[112] Chin-Sheng Yu, Yu-Ching Chen, Chih-Hao Lu, and Jenn-Kang Hwang. Prediction of protein subcellular localization. *Proteins*, 64(3):643–51, August 2006.

[113] Andea Pierleoni, Pier Luigi Martelli, Piero Fariselli, and Rita Casadio. eSLDB: eukaryotic subcellular localization database. *Nucleic Acids Research*, 35(Database issue):D208–D212, 2007.

[114] Manoj Bhasin and G P S Raghava. ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic acids research*, 32(Web Server issue):W414–9, July 2004.

[115] Sébastien Rey, Michael Acab, Jennifer L Gardy, Matthew R Laird, Katalin DeFays, Christophe Lambert, and Fiona S L Brinkman. PSORTdb: a protein subcellular localization database for bacteria. *Nucleic acids research*, 33 (Database issue):D164–8, January 2005.

[116] Aarti Garg, Manoj Bhasin, and Gajendra P S Raghava. Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search. *The Journal of biological chemistry*, 280(15):14427–32, April 2005.

[117] Robbie P Joosten, Tim a H te Beek, Elmar Krieger, Maarten L Hekkelman, Rob W W Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. A series of PDB related databases for everyday needs. *Nucleic acids research*, 39(Database issue):D411–9, January 2011.

[118] PAUL Horton, KEUN-JOON Park, TAKESHI Obayashi, and KENTA Nakai. Protein Subcellular Localization Prediction With Wolf Psort. *Proceedings of the 4th AsiaPacific Bioinformatics Conference*, 3:39–48, December 2006.

[119] Kenta Nakai and Paul Horton. Computational prediction of subcellular localization. *Methods In Molecular Biology Clifton Nj*, 390(2):429–466, 2007.

[120] B Rost, J Liu, R Nair, K O Wrzeszczynski, and Y Ofran. Automatic prediction of protein function. *Cellular and molecular life sciences : CMLS*, 60 (12):2637–50, December 2003.

[121] Man-Wai Mak, Wei Wang, and Sun-Yuan Kung. Fast subcellular localization by cascaded fusion of signal-based and homology-based methods. *Proteome science*, 9 Suppl 1:S8, January 2011.

[122] Kenichiro Imai and Kenta Nakai. Prediction of subcellular locations of proteins: where to proceed? *Proteomics*, 10(22):3970–83, November 2010.

[123] Rakesh Kaundal, Reena Saini, and Patrick X Zhao. Combining machine learning and homology-based approaches to accurately predict subcellular localization in Arabidopsis. *Plant physiology*, 154(1):36–54, September 2010.

[124] Rajesh Nair and Burkhard Rost. LOCnet and LOCtarget: sub-cellular localization for structural genomics targets. *Nucleic acids research*, 32(Web Server issue):W517–21, July 2004.

[125] Yao-Qing Shen and Gertraud Burger. Plasticity of a key metabolic pathway in fungi. *Functional & integrative genomics*, 9(2):145–51, May 2009.

[126] Mitsuteru Nakao, Roberto A Barrero, Yuri Mukai, Chie Motono, Makiko Suwa, and Kenta Nakai. Large-scale analysis of human alternative protein isoforms: pattern classification and correlation with subcellular localization signals. *Nucleic acids research*, 33(8):2355–63, January 2005.

[127] K Nishikawa, Y Kubota, and T Ooi. Classification of proteins into groups based on amino acid composition and other characters. I. Angular distribution. *Journal of biochemistry*, 94(3):981–95, September 1983.

[128] K Nishikawa, Y Kubota, and T Ooi. Classification of proteins into groups based on amino acid composition and other characters. II. Grouping into four types. *Journal of biochemistry*, 94(3):997–1007, September 1983.

[129] H Nakashima and K Nishikawa. Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. *Journal of molecular biology*, 238(1):54–61, April 1994.

[130] J Cedano, P Aloy, J a Pérez-Pons, and E Querol. Relation between amino acid composition and cellular location of proteins. *Journal of molecular biology*, 266(3):594–600, February 1997.

[131] M A Andrade, S I O'Donoghue, and Burkhard Rost. Adaptation of protein surfaces to subcellular location. *Journal of molecular biology*, 276(2):517–25, February 1998.

[132] K. Nakai and Minoru Kanehisa. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics*, 14(4):897–911, 1992.

[133] P Horton and K Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. *Proceedings of International Conference on Intelligent Systems for Molecular Biology*, 4:109–15, January 1996.

[134] Paul Horton and K Nakai. Better prediction of protein cellular localization sites with the k nearest neighbors classifier. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 5:147–52, January 1997.

[135] A Reinhardt and T Hubbard. Using neural networks for prediction of the subcellular location of proteins. *Nucleic acids research*, 26(9):2230–6, May 1998.

[136] S Hua and Z Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics (Oxford, England)*, 17(8):721–8, August 2001.

[137] Z Yuan. Prediction of protein subcellular locations using Markov chain models. *FEBS letters*, 451(1):23–6, May 1999.

[138] Tongliang Zhang, Yongsheng Ding, and Kuo-Chen Chou. Prediction of protein subcellular location using hydrophobic patterns of amino acid sequence. *Computational biology and chemistry*, 30(5):367–71, October 2006.

[139] Chin-Sheng Yu, Yu-Ching Chen, Chih-Hao Lu, and Jenn-Kang Hwang. Prediction of protein subcellular localization. *Proteins*, 64(3):643–51, August 2006.

[140] Aarti Garg and Gajendra PS Raghava. ESLpred2: improved method for predicting subcellular localization of eukaryotic proteins. *BMC Bioinformatics*, 9:503, 2008.

[141] Chittibabu Guda. pTARGET: a web server for predicting protein subcellular localization. *Nucleic Acids Research*, 34(Web Server issue):W210–W213, 2006.

[142] Hagit Shatkay, Annette Höglund, Scott Brady, Torsten Blum, Pierre Dönnes, and Oliver Kohlbacher. SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics (Oxford, England)*, 23(11):1410–7, June 2007.

[143] Sebastian Briesemeister, Torsten Blum, Scott Brady, Yin Lam, Oliver Kohlbacher, and Hagit Shatkay. SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins. *Journal of Proteome Research*, 8(11):5363–5366, 2009.

[144] Sebastian Briesemeister, Jörg Rahnenführer, and Oliver Kohlbacher. YLoc– an interpretable web server for predicting subcellular localization. *Nucleic acids research*, 38(Web Server issue):W497–502, July 2010.

[145] EBI. EMBL–European Bioinformatics Institute EMBL-EBI Annual Scientific Report 2012. Technical report, 2013.

[146] P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach.* MIT Press, Cambridge, MA, USA, 2001.

[147] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, November 1958.

[148] K Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, January 1980.

[149] G.A. Carpenter and S. Grossberg. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88, March 1988.

[150] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. October 1989.

[151] G D Stormo, Thomas D Schneider, L Gold, and A Ehrenfeucht. Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic acids research*, 10(9):2997–3011, May 1982.

[152] Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 16. Springer, New York, January 2006.

[153] G Kerr, H J Ruskin, M Crane, and P Doolan. Techniques for clustering gene expression data. *Computers in biology and medicine*, 38(3):283–93, March 2008.

[154] Lori Dalton, Virginia Ballarin, and Marcel Brun. Clustering algorithms: on learning, validation, performance, and applications to genomics. *Current genomics*, 10(6):430–45, September 2009.

[155] Markus Ringnér. What is principal component analysis? *Nature biotechnology*, 26(3):303–4, March 2008.

[156] Nils Gehlenborg, Seán I O'Donoghue, Nitin S Baliga, Alexander Goesmann, Matthew A Hibbs, Hiroaki Kitano, Oliver Kohlbacher, Heiko Neuweger, Reinhard Schneider, Dan Tenenbaum, and Anne-Claude Gavin. Visualization of omics data for systems biology. *Nature methods*, 7(3 Suppl):S56–68, March 2010.

[157] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 69:59–69, 1982.

[158] M J L de Hoon, S Imoto, J Nolan, and S Miyano. Open source clustering software. *Bioinformatics (Oxford, England)*, 20(9):1453–4, June 2004.

[159] Niklaus Fankhauser and Pascal Mäser. Identification of GPI anchor attachment signals by a Kohonen self-organizing map. *Bioinformatics (Oxford, England)*, 21(9):1846–52, May 2005.

[160] Nung Kion Lee and Dianhui Wang. SOMEA: self-organizing map based extraction algorithm for DNA motif identification with heterogeneous model. *BMC bioinformatics*, 12 Suppl 1(Suppl 1):S16, January 2011.

[161] Harris Drucker and CJC Burges. Support vector regression machines. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9: Proceedings of the 1996 Conference*, number x, pages 155–161. MIT Press, 1997.

[162] AJ Smola and B Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 2004.

[163] Debasish Basak, Srimanta Pal, and DC Patranabis. Support vector regression. *International Journal of Neural Information Processing*, 11(10):203–224, 2007.

[164] Richard G Brereton and Gavin R Lloyd. Support vector machines for classification and regression. *The Analyst*, 135(2):230–67, February 2010.

[165] V Vapnik and A Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

[166] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory - COLT 92*, pages 144–152, New York, New York, USA, 1992. ACM Press.

[167] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[168] Vladimir N. Vapnik. The nature of statistical learning theory. June 1995.

[169] B Schölkopf, A Smola, and KR Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 1998.

[170] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Information Science and Statistics. Springer New York, New York, NY, 2008.

[171] V N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks, publication of the IEEE Neural Networks Council*, 10 (5):988–99, January 1999.

[172] Olivier Chapelle, V Vapnik, O Bousquet, and S Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, pages 131–159, 2002.

[173] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 96–103, New York, New York, USA, 2008. ACM Press.

[174] Gunnar Rätsch, Takashi Onoda, and Klaus R KR Müller. Regularizing AdaBoost. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 564–570, Cambridge, MA, USA, 1999. MIT Press.

[175] K R Müller, S Mika, G Rätsch, K Tsuda, and B Schölkopf. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks,*

publication of the IEEE Neural Networks Council, 12(2):181–201, January 2001.

[176] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31:3–24, June 2007.

[177] Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: a string kernel for SVM protein classification. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 575:564–75, January 2002.

[178] Vladimir Vacic, Lilia M Iakoucheva, Stefano Lonardi, and Predrag Radivojac. Graphlet kernels for prediction of functional residues in protein structures. *Journal of computational biology : a journal of computational molecular cell biology*, 17(1):55–72, January 2010.

[179] Asa Ben-Hur, Cheng Soon Ong, Sören Sonnenburg, Bernhard Schölkopf, and Gunnar Rätsch. Support vector machines and kernels for computational biology. *PLoS computational biology*, 4(10):e1000173, October 2008.

[180] Vladimir N Vapnik. *Statistical Learning Theory*, volume 2. Wiley (New York), 1998.

[181] J Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 1999.

[182] Joseph Drish. Obtaining calibrated probability estimates from support vector machines, 2001.

[183] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, volume pp, page 694, New York, New York, USA, 2002. ACM Press.

[184] Antonis Lambrou, Harris Papadopoulos, Ilia Nouretdinov, and Alexander Gammerman. Reliable probability estimates based on support vector machines for large multiclass datasets. In *Artificial Intelligence Applications and Innovations*, pages 182–191. Springer Berlin Heidelberg, 2012.

[185] Chih-chung Chang and Chih-jen Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3): 1–27, April 2011.

[186] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3): 267–276, August 2007.

[187] Marvin Minsky and Papert Seymour. *Perceptrons.* MIT Press, 1969.

[188] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–8, April 1982.

[189] David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations.* MIT Press, January 1986.

[190] ER Kandel, JH Schwartz, and TM Jessell. *Principles of Neural Science, Fourth Edition.* McGraw-Hill Professional, 2000.

[191] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS biology*, 6(7):e159, July 2008.

[192] WS McCulloch and W Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[193] JJ Hopfield and DW Tank. Computing with neural circuits- A model. *Science*, 1986.

[194] A.K. Jain, J Mao, and K.M. Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, March 1996.

[195] Alex Graves, S Fernández, and J Schmidhuber. Multi-dimensional recurrent neural networks. In *Artificial Neural Networks - ICANN 2007*, number 1, pages 549–558. Springer-Verlag Berlin Heidelberg, 2007.

[196] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–68, May 2009.

[197] R. Rojas. *Neural Networks - A Systematic Introduction.* Springer-Verlag, Berlin, 1996.

[198] T Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78:1464–1480, 1990.

[199] Chien-Yu Huang, Long-Hui Chen, Yueh-Li Chen, and Fengming M. Chang. Evaluating the process of a genetic algorithm to improve the back-propagation network: A Monte Carlo study. *Expert Systems with Applications*, 36(2):1459–1465, 2009.

[200] Jing-Ru Zhang, Jun Zhang, Tat-Ming Lok, and Michael R. Lyu. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, 185 (2):1026–1037, February 2007.

[201] Yoshihiro Yoneda. Nucleocytoplasmic protein traffic and its significance to cell function. *Genes to cells : devoted to molecular & cellular mechanisms*, 5(10):777–87, October 2000.

[202] M Gama-Carvalho and M Carmo-Fonseca. The rules and roles of nucleocytoplasmic shuttling proteins. *FEBS letters*, 498(2-3):157–63, June 2001.

[203] UniProt Consortium. Ongoing and future developments at the Universal Protein Resource. *Nucleic acids research*, 39(Database issue):D214–9, January 2011.

[204] Neil Jentoft. Why are proteins O-glycosylated? *Trends in biochemical sciences*, 15(8):291–4, August 1990.

[205] Raj B. Parekh. Effects of glycosylation on protein function. *Current Opinion in Structural Biology*, 1(5):750–754, 1991.

[206] Ramneek Gupta and S. Brunak. Prediction of glycosylation across the human proteome and the correlation to protein function. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 310–22, January 2002.

[207] W Kabsch and C Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22 (12):2577–637, December 1983.

[208] Burkhard Rost and Chris Sander. Conservation and prediction of solvent accessibility in protein families. *Proteins*, 20(3):216–26, November 1994.

[209] S.J. Hubbard and J.M. Thornton. NACCESS, 1993.

[210] Rajmund L. Somorjai, Murray E. Alexander, Richard Baumgartner, Stephanie Booth, Christopher Bowman, Aleksander Demko, Brion Dolenko, Marina Mandelzweig, Aleksander E. Nikulin, Nicolino J. Pizzi, Erinija Pranckeviciene, Arthur R. Summers, and Peter Zhilkin. A data-driven, flexible machine learning strategy for the classification of biomedical data. In Werner Dubitzky and Francisco Azuaje, editors, *Artificial Intelligence Methods and Tools for Systems Biology*, pages 67–85. Springer Netherlands, 2004.

[211] Rafał Adamczak, Aleksey Porollo, and Jarosław Meller. Accurate prediction of solvent accessibility using neural networks-based regression. *Proteins*, 56 (4):753–67, September 2004.

[212] Rafał Adamczak, Aleksey Porollo, and Jarosław Meller. Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins*, 59(3):467–75, May 2005.

[213] J Forcier, P Bissex, and W Chun. *Python web development with Django*. Addison-Wesley, 2008.

[214] Guohui Ding, Yan Sun, Hong Li, Zhen Wang, Haiwei Fan, Chuan Wang, Dan Yang, and Yixue Li. EPGD: a comprehensive web resource for integrating and displaying eukaryotic paralog/paralogon information. *Nucleic acids research*, 36(Database issue):D255–62, January 2008.

[215] K.C. Kuo-Chen Chou and Hong-Bin H.B. Shen. Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms. *Nature protocols*, 3(2):153–162, 2008.

[216] Hong-Bin Shen and Kuo-Chen Chou. A top-down approach to enhance the power of predicting human protein subcellular localization: Hum-mPLoc 2.0. *Analytical Biochemistry*, 394(2):269–274, 2009.

[217] Pufeng Du and Chao Xu. Predicting multisite protein subcellular locations: progress and challenges. *Expert review of proteomics*, 10(3):227–37, June 2013.

[218] Annette Höglund, Pierre Dönnes, Torsten Blum, Hans-Werner Adolph, and Oliver Kohlbacher. MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics (Oxford, England)*, 22(10):1158–65, May 2006.

[219] Torsten Blum, Sebastian Briesemeister, and Oliver Kohlbacher. MultiLoc2: integrating phylogeny and Gene Ontology terms improves subcellular protein localization prediction. *BMC Bioinformatics*, 10(1):274, 2009.

[220] Robert B Best, Trevor J Rutherford, Stefan M V Freund, and Jane Clarke. Hydrophobic core fluidity of homologous protein domains: relation of side-chain dynamics to core composition and packing. *Biochemistry*, 43(5):1145–55, February 2004.

[221] Sean P Ng, Kate S Billings, Tomoo Ohashi, Mark D Allen, Robert B Best, Lucy G Randles, Harold P Erickson, and Jane Clarke. Designing an extracellular matrix protein with enhanced mechanical stability. *Proceedings of the National Academy of Sciences of the United States of America*, 104(23): 9633–7, June 2007.

[222] Mathias Uhlen. Mapping the human proteome using antibodies. *Molecular & cellular proteomics : MCP*, 6(8):1455–6, August 2007.

[223] Kevin Drew, Patrick Winters, Glenn L Butterfoss, Viktors Berstis, Keith Uplinger, Jonathan Armstrong, Michael Riffle, Erik Schweighofer, Bill Bovermann, David R Goodlett, Trisha N Davis, Dennis Shasha, Lars Malmström, and Richard Bonneau. The Proteome Folding Project: proteome-scale prediction of structure and function. *Genome research*, 21(11):1981–94, November 2011.

[224] M Maliepaard, G L Scheffer, I F Faneyte, M A van Gastelen, A C Pijnenborg, A H Schinkel, M J van De Vijver, R J Scheper, and J H Schellens. Subcellular localization and distribution of the breast cancer resistance protein transporter in normal human tissues. *Cancer research*, 61(8):3458–64, April 2001.

[225] Ying-Ying Xu, Fan Yang, Yang Zhang, and Hong-Bin Shen. An image-based multi-label human protein subcellular localization predictor (iLocator) reveals protein mislocalizations in cancer tissues. *Bioinformatics (Oxford, England)*, 29(16):2032–40, August 2013.

[226] Joel G Turner, Jana Dawson, and Daniel M Sullivan. Nuclear export of proteins and drug resistance in cancer. *Biochemical pharmacology*, 83(8): 1021–32, April 2012.

List of publications

- **A. S. Mer**, M.A. Andrade-Navarro, *A novel approach for protein subcellular location prediction using amino acid exposure.* BMC Bioinformatics. 2013 Nov 28;14(1):342.

- **A. S. Mer**, M.A. Andrade-Navarro, *Prediction of Protein Subcellular Localization using Residue Exposure* at Networks and Pathways in Bioinformatics, July 9-12, 2013 at EBI, Cambridge, UK

- **A. S. Mer**, E. G Reynaud, M.A. Andrade-Navarro, *Comparative Study Of The Evolution Of Mitochondrial And Chloroplast Genomes Respect To The Tree Of Life*, at 13th Congress of the European Society for Evolutionary Biology, 19-24 Aug, 2011 Tübingen, Germany

Appendix

Table 8.1: Comparison of NYCE to other location prediction methods. N, nuclear; Y, nucleo-cytoplasmic; C, cytoplasmic; E, extracellular; MB, membrane; ER, endoplasmic reticulum; MT, mitochondrial; CN, centriole; GA, golgi apparatus; VA, vacuolar.

Uniprot AC		Location		NYCE		Yloc		Hum-mPLoc		SherLoc		PSORT II	
Q14145	O95198	Y	C	Y	Y	C	C	MB	C	C	C	C	Y
P62714	O00743	Y	C	Y	C	C	C	C	C	C	C	C	Y
Q14145	Q9UJP4	Y	C	Y	E	C	C	MB	N	C	Y	C	C
Q5H9R7	O75170	Y	C	N	N	N	C	C	N	C	C	N	Y
Q9HC16	Q8IUX4	Y	C	N	E	N	N	N	C	C	C	C	C
O14682	Q9C0H6	Y	C	C	C	C	C	C	C	C	C	C	C
O14682	Q9NR64	Y	C	C	C	C	C	C	C	C	C	C	C
Q9Y2M5	Q9C0H6	Y	C	C	C	C	C	MB	C	C	C	ER	C
O14682	Q9UJP4	Y	C	C	E	C	C	C	N	C	Y	C	C
P42685	P51813	Y	C	Y	C	C	C	N	C	C	C	C	N
P48595	P50453	Y	C	Y	C	C	E	E	N	C	C	Y	MT
P62136	O00743	Y	C	C	C	C	C	C	C	C	C	C	Y
Q9Y2M5	Q9UJP4	Y	C	C	E	C	C	MB	N	C	Y	ER	C
O00204	O00338	Y	C	C	C	C	C	N	C	C	C	Y	C
O00204	O75897	Y	C	C	C	C	C	N	C	C	C	Y	C
O00204	Q06520	Y	C	C	C	C	C	N	C	C	C	Y	C
Q14164	Q9UHD2	Y	C	Y	C	C	C	C	C	C	C	C	C
Q9UKV8	Q9HCK5	Y	C	Y	E	Y	N	C	MB	C	C	Y	C
Q9UKV8	Q9H9G7	Y	C	Y	C	Y	N	C	MB	C	C	Y	C
Q9UKV8	Q9UL18	Y	C	Y	C	Y	Y	C	C	C	C	Y	C

Q9UM11	Q12834	Y	C	Y	C	N	N	N	Y	C	N	Y	Y
Q8ND90	Q9UL42	Y	N	C	N	PR	C	N	N	Y	C	C	C
O14862	Q6K0P9	Y	N	N	N	N	N	N	N	C	N	Y	N
P35226	P35227	Y	N	N	N	N	N	MB	N	N	N	N	N
P41218	Q6K0P9	Y	N	N	N	Y	N	N	N	N	N	N	N
Q13568	Q15306	Y	N	N	N	N	N	N	N	N	N	N	N
Q9NR64	Q9Y2M5	C	Y	C	C	C	C	C	MB	C	C	C	ER
Q86WH2	Q9NS23	C	Y	C	N	C	Y	N	N	C	C	N	N
O15182	Q8TD86	C	Y	C	E	C	C	CN	C	C	C	C	C
P26196	Q13838	C	Y	C	N	C	C	C	N	N	N	Y	C
Q12798	Q8TD86	C	Y	N	E	C	C	CN	C	C	C	C	C
Q53G59	Q14145	C	Y	Y	Y	C	C	MB	MB	C	C	C	C
Q96PQ7	Q14145	C	Y	C	Y	E	C	MB	MB	GA	C	ER	C
Q9NP86	Q8TD86	C	Y	N	E	C	C	C	C	C	C	C	C
Q9UH77	Q14145	C	Y	C	Y	C	C	N	MB	C	C	C	C
Q9Y573	Q14145	C	Y	C	Y	C	C	MB	MB	C	C	C	C
O00170	Q9NZN9	C	Y	C	N	C	C	C	C	C	C	C	C
P26196	O00148	C	Y	C	N	C	C	C	N	N	N	Y	C
P50225	O00204	C	Y	C	C	C	C	C	N	C	C	C	Y
P50226	O00204	C	Y	C	C	C	C	C	N	C	C	C	Y
O00743	P60510	C	Y	C	C	C	C	C	C	C	C	Y	C
Q9UDY6	Q86WT6	C	Y	N	C	Y	N	C	C	N	N	Y	C
O00743	P62140	C	Y	C	C	C	C	C	C	C	C	Y	C
Q9UDY6	Q6AZZ1	C	Y	N	Y	Y	Y	C	N	N	N	Y	C
Q16829	Q05923	C	N	Y	E	C	C	C	N	C	C	Y	C
Q68J44	P51452	C	N	C	N	Y	C	C	C	C	C	C	N
P15086	P15088	E	C	E	N	E	E	E	C	E	E	ER	E
P48052	P15088	E	C	C	N	E	E	MB	C	E	E	C	E
P15085	P15088	E	C	E	N	E	E	E	C	E	E	E	E
Q01196	Q13761	N	Y	N	N	N	Y	N	N	N	N	Y	N

P10828	P10826	N	Y	N	N	N	N	N	N	N	N	C	N
P43356	Q9UBF1	N	Y	N	N	C	Y	C	N	C	C	VA	Y
Q15306	Q00978	N	Y	N	N	N	N	N	N	N	N	N	MT
Q8WYH8	Q9H160	N	Y	N	N	N	N	N	N	N	N	N	N
Q9H0S4	Q13838	N	Y	C	N	N	C	MT	N	N	N	Y	C
O75676	Q15418	N	Y	Y	Y	C	C	C	N	C	C	C	C
Q5TD97	Q13642	N	Y	E	E	C	C	N	E	C	C	N	N
Q9H0S4	O00148	N	Y	C	N	N	C	MT	N	N	N	Y	C
Q01543	P11308	N	Y	N	N	N	N	N	N	N	N	N	N
Q9H3D4	O15350	N	Y	N	Y	N	N	N	N	N	N	N	N
Q96B02	P61088	N	Y	E	Y	C	C	N	N	Y	PR	MT	C
Q96B02	P68036	N	Y	E	N	C	Y	N	N	Y	PR	MT	N
Q9BTZ2	Q9BPX1	N	C	N	C	MT	Y	N	MT	PR	PR	C	C
Q13115	Q16829	N	C	N	Y	Y	C	MB	C	C	C	Y	Y