ON PREDICTING TRANSPORTER PROTEINS

By

Munira Alballa

SUBMITTED AS PHD PROPOSAL REPROT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

AT

CONCORDIA UNIVERSITY

MONTREAL, QUEBEC

NOVEMBER 2017

Contents

List of Figures

List of Tables

Glossary

Cell membrane (also called plasma membrane or plasmalemma) is biological membrane that surrounds the cytoplasm of living cells, physically separating the intracellular components from the extracellular environment.

Clade A group of all the taxa that have been derived from a common ancestor plus the common ancestor itself.

Eukaryote A eukaryote is any organism whose cells contain a nucleus and other organelles enclosed within membranes.

Homoeostasis (or Homeostasis) is the property of a system in which variables are regulated so that internal conditions remain stable and relatively constant.

Homologous The existence of shared ancestry between a pair of structures, or genes, in different species.

Hydrophilic Interacting effectively with water.

Hydrophobic Not interacting effectively with water; in general, poorly soluble or insoluble in water.

Lipids A group of naturally occurring molecules that include fats, waxes, sterols, fat-soluble vitamins (such as vitamins A, D, E, and K), monoglycerides, diglycerides, triglycerides, phospholipids, and others.

Motif A shot, functional region within protein sequence, usually recognized by sequence or structure pattern

Nonpolar A molecule or structure that lacks any net electric charge or asymmetric distribution of positive and negative charges. Nonpolar molecules generally are insoluble in water.

Polar A molecule or structure with a net electric charge or asymmetric distribution of positive and negative charges. Polar molecules are usually soluble in water.

Protein sequence The primary structure that is the unique sequence of amino acids that characterizes a given protein

Proteome The complete set of proteins expressed by an organism

R R is a programming language and environment for statistical computing and graphics.

Secondary structure The local three-dimensional structure of sheets, helices, or other forms adopted by a polynucleotide or polypeptide chain, due to electrostatic attraction between neighbouring residues.

Acronyms

AAC Amino Acid Composition

BLAST Basic Local Alignment Search Tool

HMM Hidden Markov Model

IMP Integral Membrane Proteins

IUBMB International Union of Biochemistry and Molecular Biology

MAFFT Multiple Alignment using Fast Fourier Transform

MCC Matthews Correlation Coefficient

MSA Multiple Sequence Alignment

PAAC Pair Amino Aside composition

PDB Protein Data Bank

PseAAC Pseudo-Amino Acid Composition

PSSM Position-Specific Scoring matrix

SDP Specificity-Determining Positions

T-COFFEE Tree based Consistency Objective Function for Alignment Evaluation

TCDB Transporter Classification Database

TCID Transport Classification Identification

TCS Transitive Consistency Score

TMS Transmembrane Segments

Abstract

The publication of numerous genome projects has produced an abundance of proteins sequences, many of which remain uncharacterized; transmembrane proteins are among the least characterized proteins, owing to their hydrophobic surfaces and their lack of conformational stability.

Consequently, there is an urgent need for computational approaches that use the available experimental data to distinguish and characterize transmembrane proteins. Yet, this area of research is still in its early stages, and the researchers are far from finding a definite solution. Here, we propose to characterize and predict the transport of substrates which is a major function of transmembrane proteins. In addition, we evaluate the efforts made in predicting transmembrane proteins and finding the substrate specificity of transporters. Furthermore, we conduct a preliminary study that detects the transported substrates and employs different methods including homological informations that shows promising results. We desire to further improve our method and evaluate it on large scale. Overall, we intend to implement a proteome-wide system that differentiates transporters and predicts their substrates.

Chapter I

Introduction

This chapter starts with basic biological background that are needed to understand our work. Then, Section 2 expresses motivation for working with transmembrane proteins. Followed by Section 3 that outlines the problem that we aspire to work with in the course of my PhD. Finally, Section 4 gives an overview of the rest of the report.

1 Biological background

Cell membranes are the only cellular structure found in all cells of all organisms on Earth, due to their biological significance. Membranes maintain the integrity of the cell by separating the critical chemicals and structures needed to maintain the cell from the surrounding environment. They also serve as gatekeepers, regulating the flow of molecules, energy and information in and out the cell. Furthermore, eukaryotic cells have internal membranes that enclose their organelles and control the exchange of essential cell components [1].

Cell membranes have two main components: lipids and proteins (see Figure 1) . Each component has clearly defined function: lipids form the universally conserved bilayer structure that determines membrane flexibility, basic barrier properties and how membrane proteins bind to the bilayer. At the same time, membrane proteins enable the membrane to carry out its distinctive activities with a vast diversity of cell membrane functions.

Lipids consists of two layers of phospholipid molecules whose fatty tails form the hydrophobic interior of the bilayer, and their hydrophilic polar heads line both the inside and the outside of the cell surface. Membrane proteins are embedded within the phospholipid bilayer; they come in different forms depending on cell type and subcellular location .

Some membrane proteins bind only to the membrane surface; others span the entire bilayer

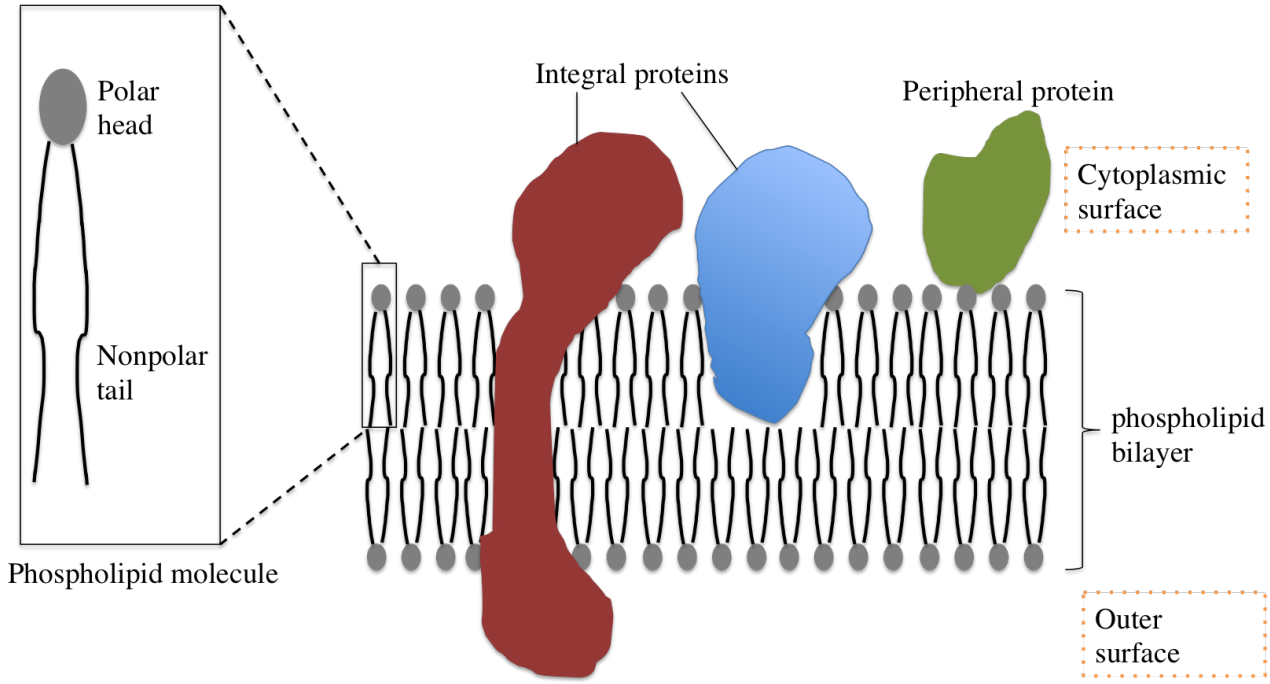

Figure 1: The structure of the cell membrane

and are exposed to water-soluble domains on both sides of the membrane (see Figure 2). The proteins buried within the bilayer are integral membrane proteins (IMPs) (also called "intrinsic proteins"). They have one or more transmembrane segments (TMSs) embedded in the bilayer in addition to extra-membranous hydrophilic segments extending into water-soluble domains on each side of the bilayer. The embedded segments are easily distinguishable because they contain residues with hydrophobic properties that interact with the nonpolar (hydrophobic) tails inside the membrane phospholipids. The IMPs are called **"transmembrane proteins"** .

Protein structures are described in four distinct levels of hierarchical organization: primary, secondary, tertiary and quaternary structures. These levels denote, respectively, the amino acid sequence of the protein, the local regular sub-structures (e.g. α-helices, β-strands), the three-dimensional structure of a single polypeptide, and the aggregation of two or more individual polypeptide chains that make up the protein complex. In this proposal, we look into the primary and secondary structures of the protein.

Membrane proteins take three general structural forms: α, β, and $\alpha\beta$-type proteins.Proteins of the α-type have TMSs formed by the connection of helices with extra-membranous loops. Such

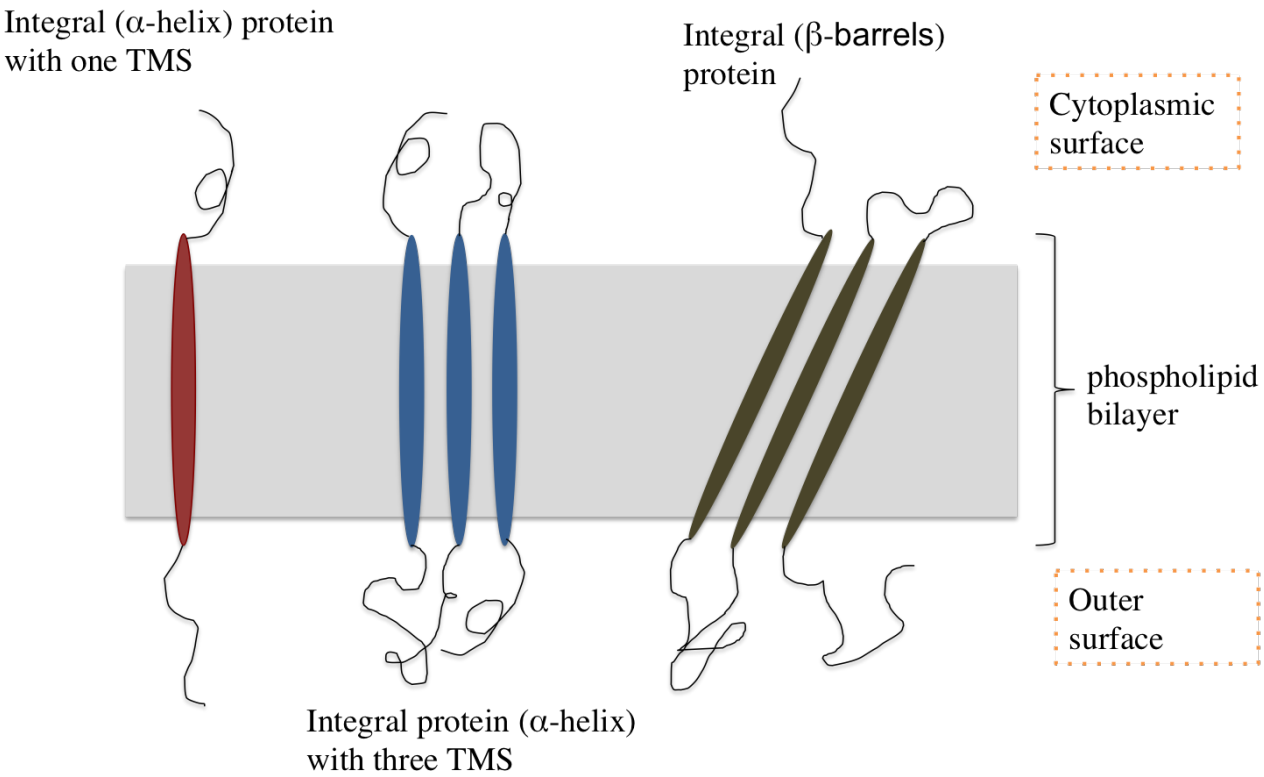

Figure 2: Schematic representation of transmembrane proteins

proteins do not have detectable β-strand structures embedded in the membrane, nor do they have extra-membranous β-strand structures. However, the majority of proteins with α-helices TMSs have extra-membranous domains that contain both α-helix and β-sheet structures. They are thus considered $\alpha\beta$-type proteins. The β-type membrane proteins are made of transmembrane β-strands that stretch across the bilayer and align in an anti-parallel fashion into large, self-enclosed β-pleated sheets with extra-membranous loops connecting adjacent β-strands. The extra-membranous loops mainly lack secondary structure (random coils), but some longer loops may contain very small α-helical regions [2].

Unlike α-helices membrane proteins that are found abundantly in all cellular membranes [3], β-barrel membrane proteins are only experimentally found in the outer membranes of Gram-negative bacteria. Some weak similarities at only the sequence level indicate the that β-barrel membrane proteins may present in the outer membrane of the mitochondria and chloroplasts [4].

1.1 Transmembrane protein classes

Categorization of membrane proteins based on their function were applied long before high-resolution structure methods become available. This makes sense because only when the functionality of such proteins is perceived can one attempt to find their underlying structure. Membrane proteins control almost all membrane functions except for the basic barrier property of the bilayer. Membrane proteins can be classified into four different functional groups [2]:

- Transporters are the membrane proteins responsible for selective permeability. They are very selective allowing certain substrates to enter or leave the cell. Channels and carriers are two major groups of transports.
- Receptors are the membrane proteins responsible for the binding of an extracellular signaling molecules and generating different intracellular signals on the opposite side of the plasma membrane.
- Enzymes are the membrane proteins responsible for various chemical reactions held in the interior surface of the plasma membrane.
- Structural proteins are the membrane proteins responsible for cell adhesion, and they carry cell surface identity marker.

It is important to mention that the functional classification of a membrane protein may not be unique. For example, some receptors form ion channels by opening or closing a channel after interaction with its ligand, and many enzymes transport substrates.

2 Motivation

Transmembrane proteins are substantial traffic gates that organize a variety of vital cellular functions including cell signaling, trafficking, metabolism and energy production. It is estimated that in an average organism one in every three proteins found in a cell is transmembrane proteins [2] [5]. For example, about 30% of the human genome is made up of transmembrane proteins. Any defected or mis-regulated membrane proteins can disturb the body's homoeostasis, giving rise to disease [6]. Therefore, the study of cell membranes is critical in understanding the causes of many diseases and finding ways to treat them. For this reason, transmembrane proteins are very attractive targets for the pharmaceutical industry; over half of today's drugs have some effect on them [7]. While the sequences of membrane proteins are known, due to the result of a number of recent genome projects, their structure and function is still not very well characterized and understood, owing to the immense effort needed to characterize them. Generally (in all proteins), experimentally finding the functions of proteins is not easy task, because the function may be related specifically to the native environment in which a particular organism lives; such an environment is hard to simulate in a lab. Particularly, membrane proteins have a hydrophobic surface, which makes extracting them from the cell membrane possible only through detergents. Also, their flexibility and instability create challenges at many levels, including crystallization, expression, and structure solution [8]. An example of how transmembrane proteins are less represented than other types of soluble proteins is in PDB (see Section 3.3). As of September 2017, less than 3% of the PDB represents the membrane proteins, with 2997 (2.1 %) α-helical structures and 899 (.6 %) β-barrel structures. Therefore, the characterization of membrane proteins and their function remains a challenge in the advancement of both structural and functional biology. Thus, It is then highly recommended to make use of the transmembrane protein sequences along with the available experimental data in computational tools to predict the transmembrane and their function. Such tools can serve as a guide to decrease the search space for experimentalists while fining the function of novel proteins. Current state-of-the-art methods remain far from a solution, but initial attempts have been made, which need further improvements.

3 Problem definition

While the amino acid sequences of many membrane proteins are available, their specific functions remain unknown. There is a consistent need for computational methods that predict the function of membrane proteins and their possible substrates. These computational methods may give a hint about the structure, function, and mechanistic features of the queried protein sequence that can be subjected to experimental verification [9].

The concepts related to transmembrane transport proteins are poorly defined. In particular, there is no single coherent problem to predicting a transport protein that all methods agree to achieve. Rather, there are different perspectives on different levels of the prediction.

Even gold standard databases are not computationally consistent. For example, TCDB [10] that offers the gold standard classification for transmembrane transporter on bases of Transporter Classification (TC) scheme, where membrane transporters classified into around 800 transporter families (for details see Chapter II. 3.2) are not consistent with Swiss-Prot [11] annotation. Proteins with the same annotations are mapped to various TCDB families (details are discussed in the preliminary study —Chapter IV)

Generally, there are two perspectives on predicting transporters: (1) based on TC family and (2) based on the substrate that the transporter transports across the membrane. The predicting based on TC family attributes a given protein to a functional family based on sequence similarity, and it does not give an accurate prediction of the transporter function. As proteins with high sequence similarity may have completely different functions, similarly, highly diverse proteins could share the same function. On the other hand, predicting the function of a given transporter and getting to the level of substrate specificity of a transporter is difficult, as it is dependent on a very small number of sites in the protein sequence, and those sites are not previously known.

Furthermore, on the level of prediction classes, there is no universally defined set of gold standard dataset. Researchers are using their own subset of substrate classes or a subset of the TC. This makes the actual problem addressed by each predictor diverse and a meaningful comparison of their performance is impossible.

During my PhD, I aspire to build a coherent proteome-wide system that can computationally detect a major function of transmembrane proteins —the transport of substrates. To accomplish this, the following research sub-questions need to be explored:

Q1: Given a protein sequence X, is this a transporter protein?

Q2: Given that protein Y is a transporter, what type of substrates does it transport across the membrane?

The issues along with the research questions will be discussed in detail in Chapter V

4 Overview

The rest of this report is organized as follows: Chapter II gives background information about different bioinformatics techniques that are used when detecting transporter proteins and their transported substrates, and goes through main transmembrane protein databases. Chapter III reviews different methods to predict transmembrane proteins topology and detect transporter proteins. Chapter IV address what we have done in transporter substrate prediction (addressing Q2). Chapter V details the limitation of the-state-of-art methods and what we aspire to accomplish during our research. Finally, Chapter VI concludes the report.

Chapter II

Background

This Chapter demonstrates important bioinformatics methods that are heavily used when addressing our research question (see Chapter I.3). Section 1 present different multiple sequence alignment (MSA) algorithms. The MSA allows us to infer homology and evolutionary relationships between different protein sequences. After that, Section 2 outlines protein composition methods that are used extensively in bioinformatics in general and substrate specificity detection methods in particular. Finally, Section 3 lists important protein databases.

1 Multiple sequence alignment

Multiple sequence alignments (MSAs) are fundamental tools for protein structure, function prediction, phylogenetic analysis, and other bioinformatics and molecular evolutionary applications.A multiple sequence alignment is a collection of more than two protein sequences that are partially or completely aligned into a rectangular array. The goal of MSA is to align the sequences in such a way that the residues in a given column are homologous in an evolutionary sense (driven from the same residue of the shared ancestry), homologous in a structural sense (occupying same positions in the three-dimensional structure), or have a common function. In closely-related sequences (40% amino acid identity or more) those three principles are essentially the same. On the other hand, if the protein sequences show some divergence over evolutionary time those principles may result in considerably different alignment and the problem of MSA becomes extremely hard to solve [12] [13]. MSA development is active an area of research; over the past decade, dozens of algorithms have been introduced. The most popular MSA algorithms will be reviewed here.

The exact methods use dynamic programing to find the global optimal alignment with time

complexity $O(L^N)$, where L is the average sequence length and N is the number of aligned sequences. Since time grows exponentially as N gets bigger, those methods are not feasible to use unless N is very small [14].

ClustalW [15], one of the most popular MSA heuristic algorithms, uses a progressive method. Firstly, the algorithm performs a pairwise alignment of all the sequences in the alignment in a matrix that shows the similarity of each pair of sequences. The similarity scores are usually converted into distance scores. Secondly, the algorithm uses the distance score matrix to construct a rough phylogenetic tree called a guide tree. Finally, ClustalW progressively aligns the sequences by following the branching order of the guide tree. Progressive methods are very efficient where hundreds of sequences can be aligned rapidly. However, when an error is introduced in the early stages in the alignment it cannot be corrected and this may increase the likelihood of misalignment due to incorrect conservation signals [13] [16].

Clustal Omega [17], the latest algorithm from the Clustal family, is highly efficient and more accurate than ClustalW. Clustal Omega is capable of aligning more than 190,000 sequences on a single processor in a matter of few hours [22]. Like ClustalW, the Clustal Omega algorithm first performs a pairwise alignment. Then, in order to reduce the number of distance calculations that are required to build the guide tree, Clustal Omega uses a modified version of mBed [18], which involves embedding the sequences in a space where the similarities within a set of sequences can be approximated without the need to compute all pair-wise distances. The sequences then can be clustered extremely quickly to produce the guide tree. Finally, progressive alignments are computed using HHalign package [19] which aligns with two hidden Markov models profiles.

Iterative methods overcome the inherited limitation of the progressive method, where the error once introduced cannot be removed. MAFFT [20] is an iterative method that uses two-cycle heuristics. Initially it aligns the sequences using progressive methods and then refines the alignment by calculating and optimizing sum-of-pairs score. MAFFT also identifies homologous regions by the fast Fourier transform where the amino acid sequence is converted to a sequence that has volume and polarity values of each amino acid residue.

The idea behind consistency-based methods is that for sequences x, y and z, if residue x_i aligns with residue y_j and y_j aligns with z_k, then x_i aligns with z_k. The consistency of each pair of residues with residue pairs from all of the other alignments is examined and weighted in such a way that reflects the degree to which those residues align consistently with other residues. T-COFFEE [21], a consistency-based method, is considered one of the most accurate available programs based on benchmarking studies. T-COFFEE takes into account both global and local pairwise alignments. Local similarity is used to reveal when two proteins share part of the sequence e.g. a domain or motif.

All of the above mentioned algorithms are general-purpose algorithms that can be used to align any related protein sequences. In other words, they use general scoring schemes that are tailored for sequences of soluble proteins. Since in transmembrane proteins the regions that are inserted into the cell membrane have a profoundly different hydrophobicity pattern compared with soluble proteins, those algorithms may not produce the optimal alignment [22].

Few packages have been published to tackle the problem of aligning transmembrane proteins, such as PROLIN-TM [22], TM-COFFEE [23] and STAM [24]. Most of these algorithms use homology extension. In homology extension methods, database searches are used to replace each sequence with the profile of closely related homologues. Consequently, each sequence position becomes a column in the multiple alignments that reveals the pattern of acceptable mutations. TM-COFFEE is the most accurate method based on benchmarking studies done by Notredame *et al.* [23]. The TM-COFFEE algorithm can be summarized as follows: for each sequence in need of alignmnet, perform a homology search using BLAST [25] and keep the hits with level of identity between 50% and 90% and a coverage of more than 70%. Then, turn the BLAST output into a profile where all columns corresponding to unaligned positions (i.e. gaps) to the query are removed and the query positions unmatched by BLAST are filled with gaps. Finally, Produce a T-COFFEE library by aligning every pair of profiles. TM-COFFEE shows a 10% improvement to the MSAProbs [26], the next best method that uses homology extension. Although homology extension methods gives much more accurate alignment, performing an alignment takes several orders of magnitude longer than the standalone applications [12].

The assessment of MSA has been the subject of research in recent years. Particularly, efforts have been devoted to answering two main questions: how to get the alignment associated with the optimal score, and how to evaluate the goodness of an alignment. A reliable way to make this evaluation is to compare the alignment result with known 3D structures as established by x-ray crystallography. Since it has been proven that even proteins with low sequence identity (less than 40%) can share a similar 3D structure, comparison of the 3D structures makes it possible to align distantly related proteins with low sequence similarity on the basis of their structural equivalence [27] [28].

Several benchmark datasets have been created as reference sets in which alignments are created from proteins having known structures. This way, one can evaluate the result of the proposed MSA algorithm on the basis of studied proteins that are experimentally and structurally homologous. Many studies devoted to comparing different MSA algorithms on tests against benchmark databases are currently available [12] [29] [30]. They can serve as a guide to researchers to choose the appropriate algorithm for a given data. The general conclusion is that there is a tradeoff between the computational cost and the accuracy; the accuracy can greatly vary if the sequences under study are highly divergent. In addition, there is no available MSA program that outperformed the others in all test cases [30]. Table 1 summarizes the advantages and disadvantages and gives general recommendations based on the recommendation of the comparative benchmarking studies.

Aligner	Advantages	Cautions	Recommendations
ClustalW	-Uses less memory than other programs -Very fast	-Less accurate than other methods	-Use when there is small number of very long sequences (more than 20,000 amino acids) -Use when aligning closely related sequences
Clustal Omega	-Fast -Accuracy is higher than ClustalW but lower than MAFFT	The performance can greatly vary on different datasets -Memory-greedy and slower than ClustalW	-Use if sequences have large N/C terminal extensions
MAFFT	-Good trade-off of accuracy and computational cost -Higher accuracy than Clustal Omega	-Requires more memory to run	-Use with sequences with large N/C terminal extensions -Use for large number of sequences (more than 500 sequences)
T-COFFEE	-Very accurate -Incorporate heterogeneous types of information	-High memory usage and execution time	-Use with 2100 sequences of typical protein length
TM-COFFEE	-The most accurate program for transmembrane protein alignment	-High computation time and memory usage on more than 100 sequences	-Use with 2100 sequences of typical protein length

Table 1: MSA programs comparison

2 Protein composition

Protein sequences have variety of information that can be used to develop a sequence based prediction method. Such information includes the amino acid compositions, the property of the amino acids such as their hydrophobicity values, hydrophilicity values, and side-chain masses. The idea of classifying proteins using amino acid composition was first introduced in 1983 by Nishikawa *et al.* [31], who found that there is a significant correlation between a protein amino acid composition and its location, such as inside the cell or outside the cell, and its functional property, such as whether the protein is an enzyme or not. Since then, amino acid composition and its different variations have been used to classify proteins according to many different properties, such as protein structure [32] [33] [34], subcellular localization [35], whether a transmembrane protein acts as a channel/pore, electrochemical potential-driven transporters, or primary active transporters [36].

In this section, formal definitions of different variations of amino acid compositions will be presented.

2.1 Amino Acid Composition (AAC)

The Amino Acid Composition is the normalized occurrence frequency of each amino acid. The fractions of all 20 natural amino acids are calculated as:

$$c_i = \frac{F_i}{L} \qquad i = (1, 2, 3, ...20) \tag{1}$$

where F_i is the frequency of the ith amino acid and L is the length of the sequence. Each protein AAC is represented as a vector of size 20:

$$AAC(P) = [c_1, c_2, c_3, ..., c_{20}] \tag{2}$$

where c_i is the composition of ith amino acid.

2.2 Pair Amino Acid Composition (PAAC)

The PAAC has an advantage over AAC since it encapsulates information about the fraction of the amino acids as well as their order. It is used to quantify the preference of amino acid residue pairs in a sequence. The PAAC is calculated as

$$d_{i,j} = \frac{F_{i,j}}{L-1} \qquad i,j = (1,2,3,...20) \tag{3}$$

where $F_{i,j}$ is the frequency of the ith and j^{th} amino acids as a pair (dipeptide) and L is the length of the sequence. Like AAC, PAAC is represented as a vector of size 400 as follows:

$$PAAC(P) = [d_{1,1}, d_{1,2}, d_{1,3}, ..., d_{20,20}] \tag{4}$$

where $d_{i,j}$ is the dipeptide composition of the ith and j^{th} amino acid.

2.3 Pseudo-Amino Acid Composition (PseAAC)

PseAAC was proposed in 2001 by Chou [37] where he showed a remarkable improvement in the prediction quality when compared to the conventional AAC. PseAAC is a combination of the 20 components of the conventional amino acid composition and a set of sequence order correlation factors that incorporates some biochemical properties. Given a protein sequence of length L:

$$R_1R_2R_3R_4...R_L \tag{5}$$

A set of descriptors called sequence order-correlated factors are defined as:

$$\begin{cases} \theta_1 = \dfrac{1}{L-1}\displaystyle\sum_{i=1}^{L-1}\Theta(R_i, R_{i+1}) \\ \theta_2 = \dfrac{1}{L-2}\displaystyle\sum_{i=1}^{L-2}\Theta(R_i, R_{i+2}) \\ \theta_3 = \dfrac{1}{L-3}\displaystyle\sum_{i=1}^{L-3}\Theta(R_i, R_{i+3}) \\ \quad . \\ \quad . \\ \quad . \\ \theta_\lambda = \dfrac{1}{L-\lambda}\displaystyle\sum_{i=1}^{L-\lambda}\Theta(R_i, R_{i+\lambda}) \end{cases} \tag{6}$$

The parameter λ is chosen such that $(\lambda < L)$. A correlation function is given by:

$$\Theta(R_i, R_j) = \frac{1}{3}\left\{[H_1(R_j) - H_1(R_i)]^2 + [H_2(R_j) - H_2(R_i)]^2 \right. \\ \left. +[M(R_j) - M(R_i)]^2\right\} \tag{7}$$

where $H_1(R)$ is the hydrophobicity value, $H_2(R)$ is hydrophilicity value, and $M(R)$ is side chain mass of the amino acid R_i. Those quantities were converted from the original hydrophobicity value, original hydrophilicity and original side chain mass by standard conversion as follows:

$$H_1(R_i) = \frac{H_1^\circ(R_i) - \frac{1}{20}\sum_{k=1}^{20} H_1^\circ(R_k)}{\sqrt{\frac{\sum_{y=1}^{20}\left[H_1^\circ(R_y) - \frac{1}{20}\sum_{k=1}^{20} H_1^\circ(R_k)\right]^2}{20}}} \tag{8}$$

where $H_1^\circ(R_i)$ is the original hydrophobicity value for the amino acid R_i that was taken from Tanford [38] ; $H_2^\circ(R_i)$ and $M^\circ(R_i)$ are converted to $H_2(R_i)$ and $M(R_i)$ in the same way. The original hydrophilicity value $H_2^\circ(R_i)$ for the amino acid R_i was taken from Hopp and Woods [39]. The mass $M^\circ(R_i)$ of the R_i amino acid side chain can be obtained from any biochemistry text book. PseAAC is represented as vector of size $(20 + \lambda)$ as follows:

$$PseAAC(P) = [s_1, ..., s_{20}, s_{21}, ..., s_{20+\lambda}] \tag{9}$$

where s_i is the pseudo-amino acid composition such that:

$$s_i = \begin{cases} \frac{f_i}{\sum_{r=1}^{20} f_r + \omega \sum_{j=1}^{\lambda} \theta_j} & 1 \leq i \leq 20 \\ \\ \frac{\omega\theta_{i-20}}{\sum_{r=1}^{20} f_r + \omega \sum_{j=1}^{\lambda} \theta_j} & 20 < i \leq 20 + \lambda \end{cases} \tag{10}$$

where f_i is the normalized occurrence frequency of the of the ith amino acid in the protein sequence, θ_j is the jth sequence order-correlated factor calculated from Equation 6, and ω is a

weight factor for the sequence order effect. The weight factor ω puts weight on the additional PseAAC components with respect to the conventional AAC components. The user can select any value from 0.05 to 0.7 for the weight factor. The default value given by Chou [37] is .05.

3 Databases

3.1 UniProt

UniProtKB (UniProt Knowledgebase) [11] is the worldwide primary database of protein sequence and functional information, and it consists of two sections, UniProtKB/Swiss-Prot and UniProtKB/TrEMBL. UniProtKB/Swiss-Prot contains well-annotated non-redundant proteins that have been manually inspected. The protein sequences in UniProtKB/Swiss-Prot are accompanied by peer-reviewed references, secondary structure elements, cross- references to other biological databases and information about their function. UniProtKB /TrEMBL contains protein sequences that are unrevised and automatically annotated. As of September 2017, UniProtKB/Swiss-Prot contains 555,426 sequence entries and UniProtKB/TrEMBL contains 89,396,316 sequence entries

3.2 Transporter Classification Database

The TCDB [10] uses the classification system approved by the International Union of Biochemistry and Molecular Biology (IUBMB) for membrane transport proteins, known as the transporter classification (TC) system. The TCDB is a curated database of accurate and experimentally characterized information from over 10,000 published references. As of September 2017, it contains more than 10,000 unique protein sequences that are classified into more than 800 transporter families. Each entry in the database has a Transport Classification Identifier (TCID) that consists of five components: V.W.X.Y.Z. where V is a number from 1-9 that corresponds to the transporter class (e.g. channels, carrier, pumps (active transport), W is a letter that refers to a transporter subclass, X is a number that refers to the transporter family, Y is also a number that corresponds to transporter subfamily and Z refers to the substrate or range of substrates transported Figure 3 exemplifies a TCDB entry.

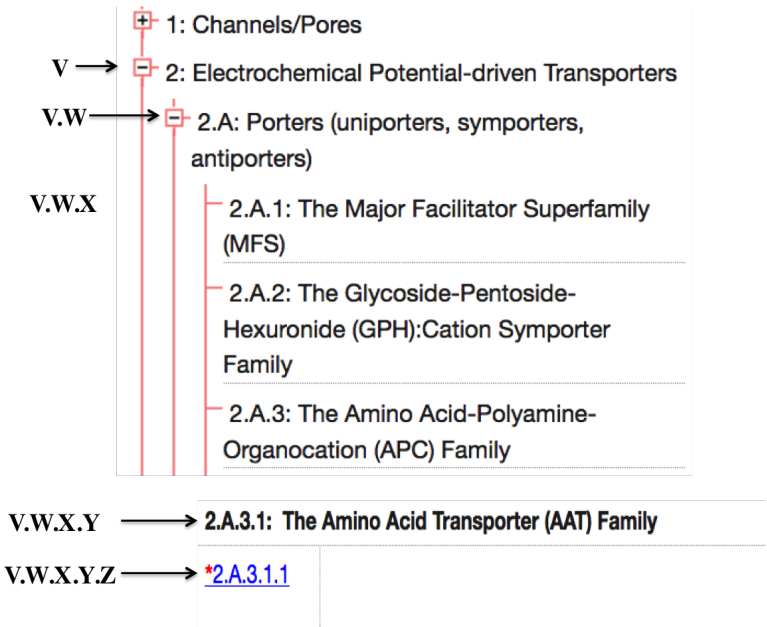

Figure 3: TCDB entry example

TCID consists of five components: V.W.X.Y.Z V is a number from 1-9 that corresponds to the transporter class (e.g. channels, carrier, pumps (active transport), W is a letter that refers to a transporter subclass, X is a number that refers to the transporter family, Y is also a number that corresponds to transporter subfamily and Z refers to the substrate or range of substrates.

3.3 Protein Data Bank

The Protein Data Bank (PDB) is single worldwide repository of information about the three-dimensional structural data of large biological molecules, such as proteins and nucleic acid. The data is typically obtained by high-resolution structure methods such as x-ray crystallography, NMR spectroscopy. As of September 2017, PDB contains 137,572 3-dimensional protein structures.

Chapter III

Literature review

This Chapter overviews what have been done to detect transmembrane proteins. Section 1 introduces different topology prediction methods used in detecting transmembrane proteins. Then, Section 2 reveals two ways to classify transporter proteins and focuses substrates specificity detection methods.

1 Transmembrane topology prediction

Transmembrane topology prediction methods predict the number of TMSs and their position in the primary protein sequence. As mentioned in Section 1, transmembrane proteins are the IMPs that span the lipid bilayer and have exposed portions on both sides of the membrane. It is expected that the portions that span the membrane contain nonpolar, hydrophobic amino acids while the portions that are in either side of the membrane consist mostly of hydrophilic, polar amino acids. The TMS can have either α-helical or β-barrel structure, The prediction methods can be classified into α-helices prediction methods and β-barrels prediction methods.

Earlier prediction methods depended solely on simple measurements like the hydrophobicity of the amino acids cite [40]. Major improvement were made after "positive- inside rule" [41] that was introduced by von Heijne in 1984, which came from the observation that positively-charged amino acids such as arginine and lysine tend to appear in the in the cytoplasmic side of the bilayer. Current methods combine hydrophobicity analysis and positive-inside rule together with machine-learning techniques and evolutionary information.

For example, MEMSATSVM [42], introduced in 2009, uses four support vector machines (SVMs) to predict transmembrane-helices, inside and outside loops, reentrant helices and signal peptides. In addition, it includes evolutionary information of many homologous protein

sequences in form of sequence profile. This method outputs predicted topologies ranked by overall likelihood, signal peptide, re-entrant helix. The reported accuracy is 89% for the correct topology and location of TM helices and 95% for correct number of TM helices. However, recent studies on experimental data [43] [44] sets reported that MEMSATSVM did not perform as well when evaluated using different datasets.

State-of-the-art methods use consensus algorithms that combine the outputs from different predictors. The highest reported prediction accuracy was achieved by TOPCONS2 [44], which is an improvement of its predecessor TOPCONS [45]. The TOPCONS2 method can more successfully distinguish between globular and membrane proteins. In addition, it is highly efficient, making it ideal to work with proteome-wide analysis . The TOPCONS2 method combine the outputs from different predictors that can also predict signal peptides -namely Philius [46], PolyPhobius [47] and SPOCTOPUS [48], OCTOPUS [49] and SCAMPI [50] into a topology profile where each residue is represented by four values: signal peptide (S), a membrane region (M) or the membrane-inside (I) and outside (O). Then, the hidden Markov model is used to process the resulting profile and predict the final topology that has the highest scoring state path.

In regard to β-barrel membrane proteins prediction, a variety of methods were introduced, such as methods that combine statistical propensities [51], k-nearest neighbor methods [52], neural networks [53] [54], hidden Markov models [55] [56] [57] [58], SVMs [59], amino acid compositions [60] [61]. Approaches based on hidden Markov model were found to have statistically significant performance when compared to other types of machine learning techniques [62]. Major players for detecting β-barrel outer membrane proteins are HHomp [63],BOCTOPUS [57], and PRED-TMBB2 [58] with reported Matthews correlation coefficient (MCC) when applied to the same dataset of (.98, .93, .92), respectively. The BOCTOPUS and HHomp techniques are much slower than PRED-TMBB2 [58]. Table 2 summarizes the prominent membrane topology predictors.

Predictor	Class	Features	Performance*	Notes
HMMTOP [64]	α-helix	-Subject to false positive predictions -Highly efficient -Does not detect signal peptide -Does not use evolutionary information -Requires a license to use the server and download	topology prediction accuracy of 70%	Highly accurate in predicting the number of TMS on accuracy of 77.1
TMHMM [65]	α-helix	-Most selective method for avoiding false positive predictions -Does not perform very well detecting the number of TMS -Does not detect signal peptide -Does not use evolutionary information -Highly efficient -freely available as web server -Requires a license for download	topology prediction accuracy of 77%	Highly efficient in Genome scale filtering of non α-helices membrane proteins [66]
MEMSAT–SVM [42]	α-helix	-Uses evolutionary information -Detects signal peptide -The performance can vary depending on the used dataset -Freely available as web server and source code	topology prediction accuracy of 89%	
TOPCONS2 [44]	α-helix	-Uses evolutionary information -Detects signal peptide -Includes consensus from five different algorithm. -Highest recorded accuracy -Freely available as web server and source code	topology prediction accuracy of 80%	Highly reliable on detecting the number, position and signal peptide in transmembrane proteins

HHomp [63]	β-barrel	-Uses evolutionary information -Extremely accurate but slow	True positives detection rate of 63.5%	
BOCTOPUS [57]	β-barrel	-Uses evolutionary information -Extremely accurate but slow	topology prediction accuracy of 83%	-Use with 2100 sequences of typical protein length
PRED-TMBB2 [58]	β-barrel	-Uses evolutionary information for but also performs well without evolutionary information when detecting β-barrel membrane proteins -Extremely efficient with performance comparable to other predictors that use evolutionary information	topology prediction accuracy of 76% MCC .92	Highly efficient in Genome scale filtering of non β-barrel membrane proteins

Table 2: Transmembrane topology prediction methods

* Performance here is the same as the reported performance by each method. Different datasets were used in the evaluation; so, proper comparison is not valid.

2 Transport proteins characterization methods

Membrane transport proteins move the hydrophilic substrates across the hydrophilic membrane between cell compartments or between different cells. Knowledge of the substrate specificities of a transporter is a necessity to fully perceive its role and it is important information for the annotation of transport proteins. Generally, a transporter protein can be classified on the basis of transporter family or according to the substrate it transports [67].

2.1 TC-family classification

The classification into families commonly follows the TCDB system (see Chapter II. 3.2). Many of the earlier bioinformatics efforts classified transporter proteins to their corresponding putative families by using MSAs and phylogeny [68] [9]. The rationale behind using those techniques is that proteins with high sequence similarity are typically homologous and thus belong to the same family. This may give a hint about the structure, function, and mechanistic features of the queried protein sequence that can be subjected to experimental verification [9]. More advanced methods that incorporate machine-learning methods were also used to predict TCDB family. For example, TransportTP [10] classifies a transporter to TCDB families in two phases: the first phase uses traditional homology methods to predict the queried transporter based on sequence similarity to the classified proteins in TCDB. The second phase employs machine-learning methods to refine the initial prediction by collecting different features such as TMSs and the top-k nearest neighbors in TCDB, homologs in Pfam and Gene Ontology, and non-transporter homologs from Swiss-Prot. The main limitation of classifying a transporter based on TCDB family, however, is that homologous sequences do not always share significant sequence similarity. Likewise, proteins with high sequence similarity do not always share the same function [69]. Therefore, it is often irrational to predict the transported substrate based on these methods, because two proteins that transport the same substrate may belong to different families; likewise, transporters belonging to the same family may transport different substrates.

2.2 Substrate specificity classification

The studies that classify a transporter protein according to the substrate it transports are quite limited. The highest reported accuracy was reported by **Schaadt *et al.* [70] in 2010**, where researchers used an amino acid composition (AAC), pair amino acid composition (PAAC), and pseudo-amino acid composition methods (PseAAC) (see Chapter II.2) in addition to amino acid conservation with homologues sequences, called MSA-AAC, to detect different substrate specificity.

The MSA-AAC method uses a full multiple sequence alignment (MSA) of each protein in the dataset built by ClustalW [15] For this, homologous sequences was searched in the nonredundant database using BLAST. Then, Sequences with an identity below 25% were removed. The occurrence of every amino acid in all sequences of the alignment was normalized by the numbers of included amino acids and the resulted vector of size 20 was considerd.

The investigations were done on Arabidopsis Thaliana transmembrane proteins, and they considered four different substrates classes: amino acid, oligopeptides, phosphate and hexose with a total of 61 transporters in the positive data set. This method relies on the Euclidean distance between the query protein sequence composition and the mean composition of protein sequences of each substrate class to compute a score for each query sequence against each substrate class. Their approach has a high accuracy around 90%, compared to 60% for randomized data. Although the performance is promising, their data set contains limited transporters of only one organism.

In 2011, Chen *et al.* [71] utilized AAC, PAAC, and biochemical properties using the AAindex database [72] along with some evolutionary information in form of position-specific scoring matrices (PSSMs) to classify a transporter to four substrate classes: electron, protein/mRNA, ion and others. Their dataset is not tailored to a specific organism and contains a total of 651 transporters. A neural network was employed to construct the classifier. The method produced an accuracy of about 80%,

In 2012, Schaadt *et al.* [73] found that separating TMSs and non-TMSs when calculating amino acid compositions yields to an improved accuracy of 80% in comparison to 76% when the

composition is computed for the whole sequence. This method also used Arabidopsis Thaliana transmembrane proteins considered the same four substrates classes: amino acid, oligopeptides, phosphate and hexose, with a total of 61 transporters.

In 2013, Barghash *et al.* [74] applied three different approaches: BLAST [25], which generates alignments that optimize a measure of local similarity, HMMER [75] which searches sequence databases for sequence homologs using probabilistic methods and MEME [76] which discovers motifs in protein sequences using expectation maximization. These methods, under different thresholds, were used to evaluate whether annotations about the transporter substrate could be transferred from one organism to the other. Four substrates classes were considered: metal ions, phosphate, sugar, and amino acid transporters from Escherichia Coli (72 transporters), Saccharomyces Cerevisiae (79 transporters), and Arabidopsis Thaliana (95 transports). They found that in the use of these methods, sequences tend to match sequences from their TC families rather than sequences in the same substrate family. Their reported performance was low for substrate-level classification with an F-measure around 40-75%.

In 2014, Mishra *et al.* [77] developed a web server, TrSSP, for predicting the substrate specificity of transporters. Protein sequence features such as AAC, PAAC, physico-chemical composition, biochemical composition AAindex database and position-specific scoring matrices (PSSM) were used to predict the substrate specificity of seven transporter classes: amino acid, anion, cation, electron, protein/mRNA, sugar, and other transporters. Biochemical composition was computed using a set of 49 selected physical, chemical, energetic, and conformational properties to define the biochemical composition of each protein sequence. The 49 values were selected from the AAIndex database [78] and were successfully applied in many areas in bioinformatics, such as protein folding, transporter classification [71]. The normalized values for 49 Amino acid properties were used. The normalized values along with a brief description of each value is available from: `https://www.iitm.ac.in/bioinfo/fold_rate/` Then, for each protein, a vector size of 49 that represents the biochemical composition was computed as follows:

$$AAindex_i = \frac{\sum_{j=1}^{n} AAindex_{ij}}{n} \tag{11}$$

where $AAindex_i$ is the value of the ith biochemical property (a total of 49 properties) and n is

the sequence length.

The PSSM is constructed with PSI-BLAST [23] which uses BLAST to build a PSSM from the multiple alignments of the highest scoring hits in an initial BLAST search with default threshold $e - value = 1e - 3$. Then, this PSSM is used again to search the database for new matches; the newly detected sequences are incorporated to update and refine the PSSM profile in every iteration. PSI-BLAST returns a PSSM that has 20 rows (one for each amino acid) and n columns (a column for each position in the queried amino acid sequence). The value in each cell represents the probability occurrence of amino acid i in position j. Then, because the number of columns in the PSSM is depends on the length of the queried sequence (n) we need to fix the size for all sequences. To do so, the number of column is reduced to 20 by summing all the rows in the PSSM that correspond to the same amino acid in the primary sequence. Finally, the values are divided by the length of the sequence and normalized to a range of $0 - 1$ using the following general formula:

$$V' = \frac{V - min}{max - min} \tag{12}$$

where V' is the normalized value, V is the value of PSSM after the sum and division, min is the minimum value in PSSM and max is the maximum value. result is a vector of size of 400, which represents the PSSM for each sequence.

Finally, an SVM was applied for classification. Their method found that the best performance was achieved by combining biochemical composition and PSSM with an overall Mathews correlation coefficient (MCC) of 0.41.

A summary of the proposed solutions can be found in Table 3

Solution	Organism	Size	Substrates	Features	Classifier	Performance*
Schaadt *et al.* [70]	Specific (Arabidopsis thaliana)	61	amino acid, oligopeptides, phosphate and hexose	AAC, PAAC, PseAAC, MSA-AAC	Euclidean distance	Accuracy of 90%
Chen *et al.* [71]	General	651	electron, protein/ mRNA, ion and others	AAC, PAAC, AAindex, PSSM	Neural network	Accuracy of about 80%
Schaadt *et al.* [73]	Specific (Arabidopsis thaliana)	61	amino acid, oligopeptides, phosphate and hexose	AAC with separating TM-segments	Euclidean distance	Accuracy of 80%
Barghash *et al.* [74]	Specific (Escherichia coli, Saccharomyces cerevisiae, Arabidopsis thaliana)	246	amino acids, metal ions, phosphates and sugars	BLAST, HMMER, MEME	N/A	F-measure around 40-75%
Mishra *et al.* [77]	General	780	amino acid, anion, cation, electron, protein/ mRNA, sugar and others	AAC, PAAC, PseAAC, AAindex, PSSM	SVM	Overall MCC of 0.41 and accuracy of 78%

Table 3: Substrate specificity solutions

* Performance here is the same as the reported performance by each method; different methods use different datasets with various prediction classes. So, proper evaluation is difficult

Chapter IV

Preliminary study

Here we investigate the different techniques used in classifying transmembrane proteins based on their transported substrate. We used Mishra *et al.* [77] paper as our main reference in terms of data and performance, since their work is the latest published work that focuses on substrate specificity that also claim to outperform other classifiers. In addition, we have developed and assisted new techniques.

1 Materials and methods

1.1 Datasets

We used the main dataset and testing dataset from Mishra *et al.* [77]. (available at: `http://bioinfo.noble.org/TrSSP`). Their data (Table 4) was collected from the Swiss-Prot (release 2013-03) database, and manually curated. Seven total substrate classes were considered: amino acid, anion, cation, electron, protein/mRNA, sugar, and others. Others refer to transporters that does not belong to any of the other six classes. A total of 760 transporters for the main dataset and 120 for the testing dataset were used.

Transporter class	Main dataset	Testing dataset
Amino acid	70	15
Anion	60	12
Electron	60	10
Cation	260	36
Protein/mRNA	70	15
Sugar	60	12
Other	200	20
Total transporters	**780**	**120**

Table 4: The numbers of samples in the main dataset and testing dataset for seven transporter classes

Further, we map all of the transporters to TCDB families. To perform the mapping, a local TCDB database was created. Then, homology search using BLAST was performed on each sequence in the dataset. A transporter is mapped to its corresponding TCDB family if there is an exact match or the normalized $e - value$ is below $1e - 8$. This threshold was suggested by Barghash *et al.* [74] as an acceptable threshold on BLAST when dealing with a TC system.

1.2 Protein sequence encoding

we implemented **AAC, PAAC, PseAAC, AAindex, PSSM**, in same way Mishra *et al.* [77] described in their work (see Chapter III. 2). Further, we also used Schaadt *et al.* [70] method (MSA-AAC) on substrate specificity prediction. We made few modifications to MSA-AAC method, we call our method **filtered-MSA-AAC**. First, instead of non-redundant database, we created a local database by combining all sequences from Swiss-Prot that have *transmembrane* keyword. The size of this database is only 40 MB in comparison to 18 GB for the non-redundant database. Second, instead of retrieving a 1,000 sequences we limited the number of the retrieved sequences to 120 to fit our computational power in the alignment phase. Next, we aligned the sequence using TM-COFFEE (Version-11.00.8cbe486) instead of ClustalW. with the following command:

```
t_coffee mysequences.fasta -mode psicoffee -protein_db uniref50-TM
-template_file PSITM
```

Where *mysequences.fasta* contains the sequences of 120 homologous retrieved from BLAST. Third, we applied transitive consistency score (TCS) [79] to the alignment. The TCS is a scoring scheme that uses a consistency transformation to assign a reliability index to every pair of aligned residues, to each individual residue in the alignment, to each column, and to the overall alignment. The reliably index ranges from 0 to 9, where 0 is extremely uncertain and 9 is very reliable, sample of TCS output is presented in Figure 4. When applying TCS score to the alignment, we were able to filter around 50% of sequences length.

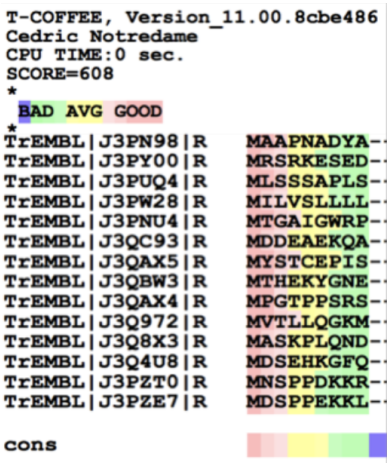

Figure 4: Sample of TCS scoring output

Columns with a reliability index of below 3 were removed using the following command:

```
t_coffee -infile myMSA.aln -evaluate -output tcs_column_filter3_fasta
```

where *myMSA.aln* is the MSA file, *tcs_column_filter3_fasta* is the filtered file in FASTA format. Finally using the filtered file we computed the AAC.

Further, we implemented **filtered-MSA-PAAC** by using the filtered MSA file and calculating the average occurrence of every amino acid residue pairs in all sequences of the alignment, we got a vector of size 400 as a result.

We also implemented **filtered-MSA-PseAAC** using the filtered MSA file and calculating the average occurrence of every amino acid residue in all sequences of the alignment (total of 20) in addition to the average sequence order correlation factors ($\lambda = 30$) among all sequences in the alignment, we got a vector of size 50 as a result.

Lastly, we incorporated a new feature, we call it **Filtered-HMM-Profile**. the HMM-profile

are used to model MSA, where it effectively represent the common patterns in the alignment. using HMMER package [80] (Version HMMER 3.1b2), we first build HMMs database (see Figure 5). To do this, MSA was built and then filtered for each sequence in the dataset. This filtered MSA was used to build HMM profile using *hmmbuild* for each sequence in the dataset. All of the HMM profiles were combined into HMMs database. Then, to predict the query

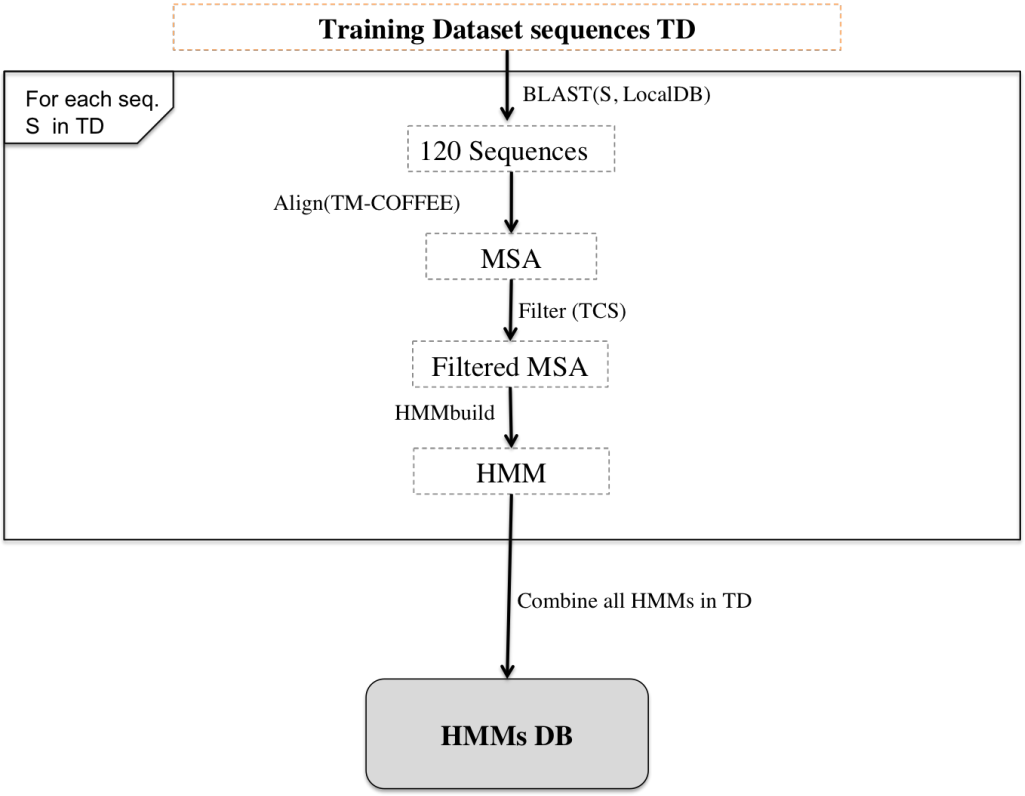

Figure 5: Filtered-HMM-Profile database building process

Each sequence S in the training dataset went through the execution pipeline. First, we perform homology search using BLAST for S against the local Swiss-Prot database. Then, we used the retrieved sequences to build MSA using TM-COFFEE, after that we filter the MSA using TCS. The filtered MSA is then used to build HMM using HMMER's *hmmbuild*. Finally, all HMMs are combined into HMMs database.

protein class on **Filtered-HMM-Profile**, *hmmscan* is used to scan query protein sequences against HMMs database. We predict the query protein class to be the same as the highest scoring hit (see Figure 6)

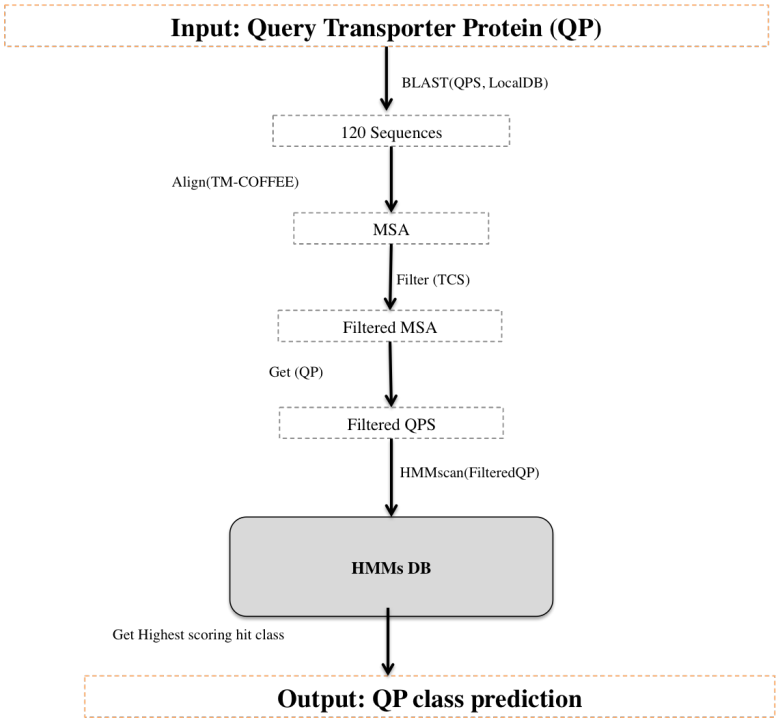

Figure 6: Filtered-HMM-Profile class prediction process

To predict query protein sequence (QP) class, we first search BLAST for homologous sequences to QP against the local Swiss-Prot database . Then, we used the retrieved sequences to build MSA using TM-COFFEE, after that we filter the MSA using TCS. Then from The filtered MSA we retrieve the filtered QP. Finally we scan the filtered QP against out HMMs database using HMMER's *hmmscan*. Finally, we predict the QP class as the same class as the highest scoring hit class

1.3 Classification

For the rest of the features, we use Support Vector Machine (SVM) with RBF kernel as implemented by R e1071 library version 1.6-8. Since we have seven different classes, we used multi-class SMV that is implemented by e1071 using a one-against-one approach, in which $(7 \times 6)/2 = 21$ binary classifiers are trained; the predicted class is found through a voting scheme where all the binary classifiers are applied, the class that gets highest number of votes is predicted by the combined classifier.

1.4 Performance measurement

Four statistical measures were considered to measure the performance. Sensitivity, which calculates the proportion of positives that are correctly identified.

$$Sensitivity = \frac{TP}{TP + FN} \tag{13}$$

Specificity that measures the the proportion of negatives that are correctly identified.

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

Accuracy that proportion of correct predictions made divided by the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{15}$$

Matthews correlation coefficient (MCC) is less influenced by imbalanced test because it takes into account true and false positives and negatives, MCC values range from 1 to -1 where 1 indicates a perfect prediction, 0 represent no better than random and -1 implies total disagreement between prediction and observation. higher MCC value means the predictor has high accuracies on positive and negative classes, and also less misclassification on the two classes. MCC is argued to be the best singular assessment metric specially when the data is imbalanced [81] [82] [83].

$$MCC = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{16}$$

The calculation of the MCC in the multiclass case was reported originally in [84]. This measure is called R_K statistics, and is calculated by $K \times K$ confusion matrix C as follows:

$$\text{MCC} = \frac{\sum_k \sum_l \sum_m C_{kk} C_{lm} - C_{kl} C_{mk}}{\sqrt{\sum_k (\sum_l C_{kl})(\sum_{k'|k' \neq k} \sum_{l'} C_{k'l'})} \sqrt{\sum_k (\sum_l C_{lk})(\sum_{k'|k' \neq k} \sum_{l'} C_{l'k'})}} \tag{17}$$

2 Experiments

In this section. we present the main experiments that we conducted. To evaluate the performance of different models, five-fold cross-validation was applied. Where the dataset is randomly partitioned into five equal sized subsamples. A single subsample is kept as the validation data and the remaining four subsamples are used to train the model. This model is then tested using the retained subsample. The cross-validation process is repeated four times where each of the subsamples is used once as the validation data. The performance of each model is averaged to produce a single estimation.

2.1 Protein compostions

Here, we examine the use of amino acid features without any incorporation of homology information. first AAC five-fold cross-validation is as follows:

Class	**Specificity**	**Sensitivity**	**Accuracy**	**MCC**
Amino acid	95.22	37.00	83.28	0.34
Anion	98.76	4.61	84.20	0.02
Cation	59.79	74.83	56.35	0.17
Electron	99.45	20.16	86.81	0.36
Protein	97.90	27.13	84.32	0.32
Sugar	98.05	36.59	86.53	0.34
Other	75.90	43.37	58.30	0.09
Overall			45.38	0.22

Table 5: AAC cross-validation performance

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	96.92	42.86	84.77	0.42
Anion	98.61	14.51	84.95	0.19
Cation	59.47	69.24	54.78	0.14
Electron	99.45	11.18	85.98	0.22
Protein	97.33	25.33	83.39	0.26
Sugar	99.03	37.55	88.58	0.48
Other	73.34	43.53	56.72	0.06
Overall			45.89	0.25

Table 7: PAAC cross-validation performance

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	96.92	48.21	86.55	0.47
Anion	97.23	10.88	83.48	0.08
Cation	58.07	79.70	57.93	0.24
Electron	99.03	8.87	85.46	0.13
Protein	99.44	21.50	86.32	0.36
Sugar	97.65	31.74	86.47	0.34
Other	79.43	42.61	61.27	0.13
Overall			47.69	0.27

Table 9: PseAAC cross-validation performance

Class	**Specificity**	**Sensitivity**	**Accuracy**	**MCC**
amino acid	99.16	3.25	78.63	0.00
anion	100.00	0.00	82.04	0.00
cation	36.33	80.43	41.66	0.36
electron	99.73	8.89	82.90	-0.04
protein	99.02	4.12	78.57	0.20
sugar	100.00	0.00	82.00	-0.15
other	72.16	27.14	46.89	0.06
Overall			33.46	0.01

Table 11: AAindex cross-validation performance

2.2 Protein compostions with homology information

the cross-validation performance when homology information in form of MSA without any filtering is incorporated is presented here.

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	97.46	74.78	93.11	0.70
Anion	97.23	23.25	88.06	0.24
Cation	82.76	73.28	75.28	0.50
Electron	97.51	54.36	91.35	0.54
Protein	95.65	52.11	88.40	0.38
Sugar	81.57	62.16	72.70	0.72
Other	98.88	65.23	94.40	0.47
Overall			61.28	0.49

Table 13: MSA-AAC cross-validation performance

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	97.46	74.78	93.11	0.70
Anion	97.23	23.25	88.06	0.24
Cation	82.76	73.28	75.28	0.50
Electron	97.51	54.36	91.35	0.54
Protein	95.65	52.11	88.40	0.38
Sugar	81.57	62.16	72.70	0.72
Other	98.88	65.23	94.40	0.47
Overall			63.92	0.51

Table 15: MSA-PAAC cross-validation performance

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	98.03	72.23	93.52	0.71
Anion	98.47	33.43	90.35	0.42
Cation	78.15	76.04	73.07	0.47
Electron	97.23	37.36	89.52	0.40
Protein	97.88	43.92	89.83	0.49
Sugar	98.75	70.00	94.64	0.73
Other	79.37	56.34	69.93	0.31
Overall			61.79	0.50

Table 17: MSA-PseAAC cross-validation performance

2.3 Protein compostions with filtered homology information

Protein compositions with Homology information in form of MSA and TCS filtering is presented here.

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	97.04	76.99	93.02	0.70
Anion	96.80	39.79	89.48	0.39
Cation	82.64	77.28	77.21	0.54
Electron	97.38	48.24	91.03	0.50
Protein	97.33	50.89	90.33	0.52
Sugar	97.65	79.62	94.62	0.74
Other	85.36	58.00	75.14	0.41
Overall			66.92	0.56

Table 19: filtered-MSA-AAC cross-validation performance

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	97.57	71.76	93.53	0.71
Anion	98.89	32.36	91.55	0.44
Cation	81.34	79.33	77.60	0.55
Electron	98.18	48.92	92.44	0.54
Protein	96.90	54.43	90.65	0.53
Sugar	98.88	72.44	95.24	0.76
Other	84.13	67.96	77.05	0.47
Overall			67.05	0.60

Table 21: filtered-MSA-PAAC cross-validation performance

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	97.60	79.06	94.16	0.74
Anion	98.19	34.26	90.34	0.42
Cation	81.72	77.24	76.31	0.52
Electron	97.22	31.86	89.29	0.33
Protein	97.61	41.68	89.68	0.45
Sugar	98.72	66.35	94.82	0.72
Other	80.64	64.59	73.06	0.40
Overall			63.71	0.51

Table 23: filtered-MSA-PseAAC cross-validation performance

2.4 Homology information in form of PSSM

The cross validation performance when homology information is incorporated in form of PSSM. Here, we do not use any form of protein compositions.

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	95.78	61.66	88.39	0.53
Anion	99.03	20.68	88.64	0.33
Cation	71.74	75.52	67.07	0.36
Electron	98.06	21.00	87.38	0.25
Protein	97.47	38.42	87.55	0.40
Sugar	98.33	51.61	91.26	0.57
Other	78.71	52.74	66.16	0.25
Overall			55.38	0.39

Table 25: PSSM cross-validation performance

2.5 Filtered-HMM-Profile

The cross validation performance when HHMs are incorporated

Class	Specificity	Sensitivity	Accuracy	MCC
Amino acid	99.15	70.75	95.23	0.77
Anion	99.03	38.49	92.37	0.51
Cation	79.26	81.45	77.44	0.56
Electron	98.34	49.06	92.50	0.56
Protein	97.33	59.76	91.78	0.51
Sugar	99.18	75.27	96.38	0.59
Other	84.89	67.74	77.60	0.79
Overall			68.84	0.64

Table 27: filtered-HMM-Profilecross-validation performance

3 Discussion

When mapping the transporters to TCDB families, we found that 537 sequences (68.84%) have exact matches, 125 (16%) sequences have matches below the specified threshold, and the remaining sequences have no matches. Furthermore, not all transporters that transport the same substrate belong to the same TCDB family, as presented in Tables 30-33. This demonstrates that proteins with the same function may not have high sequence similarity. For example, amino acid transporters belong to 13 different families and cation transporters include members from 47 different families. Likewise, the same TCDB family can be found in more than one substrate class. For example, some transporters from amino acid, anion, cation and others share the same TCDB superfamily 1.A.2. Moreover, amino acid and others share the same subfamily 1.A.2.14. This confirms that the transporters substrates specificity are sparse in TCDB families, and using sequence similarity alone to detect the transporter function may give misleading results. Other features are thus needed to differentiate proteins with distinct functions.

TCDB Super-family	**TCDB subfamily**
1.I.1	1.I.1.1 (8)
2.A.64	2.A.64.1 (2)
3.A.1	3.A.1.113 (1) 3.A.1.125 (2)
3.A.5	3.A.5.1 (3) 3.A.5.8 (4)
3.A.8	3.A.8.1 (9)
3.A.9	3.A.9.1 (2)
3.A.16	3.A.16.1 (1)
3.A.18	3.A.18.1 (1)
3.A.20	3.A.20.1 (2)
3.A.22	3.A.22.1 (1)
9.A.14	9.A.14.1 (1)
9.A.18	9.A.18.1 (1)
9.A.62	9.A.62.1 (1)

Table 28: Protein mapping to TCDB families

TCDB Super-family	TCDB subfamily
2.A.1	2.A.1.3 (1) 2.A.1.14 (1) 2.A.1.44 (1) 2.A.1.48 (2)
2.A.3	2.A.3.2 (2) 2.A.3.3 (1) 2.A.3.8 (7) 2.A.3.10 (11)
2.A.7	2.A.7.17 (1) 2.A.76 (1)
2.A.17	2.A.17.1 (2) 2.A.17.2 (1) 2.A.17.3 (1) 2.A.17.4 (3)
2.A.18	2.A.18.2 (4) 2.A.18.5 (1) 2.A.18.6 (5) 2.A.18.7 (2) 2.A.18.8 (1)
2.A.22	2.A.22.6.2 (3)
2.A.23	2.A.23.2 (3) 2.A.23.3 (2) 2.A.23.4 (1)
2.A.26	2.A.26.1 (1)
2.A.29	2.A.29.9 (1) 2.A.29.14 (1) 2.A.29.19 (1)
2.A.42	2.A.42.2 (1)
2.A.76	2.A.76.1 (1)
3.A.1	3.A.1.3 (1) 3.A.1.5 (3)
8.A.9	8.A.9.1 (1) 8.A.9.2 (1)

Table 29: Amino acid mapping to TCDB families

TCDB Super-family	TCDB subfamily
1.A.17	1.A.17.1 (2)
1.A.46	1.A.46.1 (1)
1.B.8	1.B.8.1 (3)
2.A.1	2.A.1.15 (1) 2.A.1.19 (1) 2.A.1.8 (1) 2.A.1.9 (1)
2.A.16	2.A.16.3 (1) 2.A.16.5 (1)
2.A.17	2.A.17.3 (5)
2.A.21	2.A.21.5(1)
2.A.29	2.A.29.13(1) 2.A.29.2 (1) 2.A.29.4 (2)
2.A.31	2.A.31.1 (2)
2.A.40	2.A.40.4 (1)
2.A.47	2.A.47.1 (1) 2.A.47.2 (2)
2.A.49	2.A.49.3 (2) 2.A.49.5 (1)
2.A.53	2.A.53.1 (2) 2.A.53.2 (3)
2.A.59	2.A.59.1 (1)
2.A.60	2.A.60.1 (3)
2.A.66	2.A.66.1 (1)
3.A.1	3.A.1.9 (1) 3.A.1.202 (1) 3.A.1.208 (1)
3.A.4	3.A.4.1(1)

Table 30: Anion Dataset mapping to TCDB families

TCDB Super-family	TCDB subfamily
3.D.1	3.D.1.1 (9)
3.D.3	3.D.3.5 (1)
3.D.4	3.D.4.3 (3) 3.D.4.5 (3)
5.A.3	5.A.3.1 (2) 5.A.3.4 (1)
5.B.1	5.B.1.1 (1) 5.B.1.4 (1)

Table 31: Cation mapping to TCDB families

TCDB Super-family	TCDB subfamily
2.A.1	2.A.1.1 (21) 2.A.1.4 (1) 2.A.1.5 (1) 2.A.1.7 (1) 2.A.1.20 (1)
2.A.2	2.A.2.4.(4)
2.A.7	2.A.7.9 (1) 2.A.7.10 (2) 2.A.7.12 (1) 2.A.7.13 (2) 2.A.7.15 (2)
2.A.16	2.A.16.2 (1)
2.A.21	2.A.21.3 (4)
2.A.56	2.A.56.1 (1)
2.A.123	2.A.123.1 (1)
3.A.1	3.A.1.1 (3) 3.A.1.2 (2) 3.A.1.108 (1) 3.A.1.139 (1)
4.A.1	4.A.1.1 (1)
4.A.7	4.A.7.1 (3)
9.A.58	9.A.58.1 (1)

Sugar Dataset mapping to TCDB families

TCDB Super-family	TCDB subfamily
2.A.108	2.A.108.1 (1) 2.A.108.2 (1)
3.A.1	3.A.1.14 (1) 3.A.1.18 (2) 3.A.1.201 (1) 3.A.1.205 (2) 3.A.1.208 (4) 3.A.1.21 (2) 3.A.1.210 (1) 3.A.1.23 (5)
3.A.2	3.A.2.1 (13) 3.A.2.2 (11)
3.A.3	3.A.3.3 (1) 3.A.3.5 (3)
8.A.10	8.A.10.2 (1)
8.A.14	8.A.14.1 (3)
8.A.16	8.A.16.2 (1)
8.A.22	8.A.22.1 (5)
8.A.28	8.A.28.1 (1)
9.A.8	9.A.8.1 (1)
9.A.9	9.A.9.1 (1)
9.A.40	9.A.40.3 (1)
9.B.37	9.B.37.3 (1)

Cont. Cation mapping to TCDB families

TCDB Super-family	TCDB subfamily
1.A.1	1.A.1.2 (11) 1.A.1.3 (4) 1.A.1.4 (6) 1.A.1.5 (2) 1.A.1.6 (1) 1.A.1.7 (1) 1.A.1.8 (2) 1.A.1.9 (3) 1.A.1.10 (5) 1.A.1.11 (12) 1.A.1.13 (1) 1.A.1.15 (3) 1.A.1.16 (2) 1.A.1.18 (2) 1.A.1.19 (3) 1.A.1.20 (3)
1.A.2	1.A.2.1 (8)
1.A.4	1.A.4.4 (1)
1.A.5	1.A.5.2 (1) 1.A.5.3 (1)
1.A.6	1.A.6.1 (10) 1.A.6.2 (2)
1.A.11	1.A.11.4 (1)
1.A.23	1.A.23.4 (1)
1.A.26	1.A.26.2 (1)
1.A.35	1.A.35.1 (1) 1.A.35.2 (2) 1.A.35.4 (1)
1.A.51	1.A.51.1 (1)
1.A.52	1.A.52.1 (1)
1.A.56	1.A.56.1 (6) 1.A.56.2 (1)
1.A.77	1.A.77.1 (1)
1.A.87	1.A.87.3 (1)
2.A.1	2.A.1.16 (1) 2.A.1.19 (3) 2.A.1.2 (1)
2.A.4	2.A.4.4 (2) 2.A.4.7 (1)
2.A.5	2.A.5.4 (2) 2.A.5.5 (1)
2.A.6	2.A.6.1 (1)
2.A.7	2.A.7.1 (1) 2.A.7.25 (1)
2.A.19	2.A.19.1 (1) 2.A.19.2 (3) 2.A.19.4 (1)
2.A.22	2.A.22.2 (1)
2.A.29	2.A.29.5 (1)
2.A.33	2.A.33.1 (6)
2.A.34	2.A.34.1 (2)
2.A.35	2.A.35.1 (1)
2.A.36	2.A.36.1 (2) 2.A.36.7 (1)
2.A.37	2.A.37.4 (3) 2.A.37.5 (1)
2.A.38	2.A.38.2 (1) 2.A.38.3 (3)
2.A.55	2.A.55.1 (2) 2.A.55.2 (4) 2.A.55.3 (1)
2.A.58	2.A.58.1 (1)
2.A.63	2.A.63.1 (5)
2.A.72	2.A.72.3 (2)
2.A.92	2.A.92.1 (1)
2.A.96	2.A.96.1 (1)

Table 34: Cation mapping to TCDB families

TCDB Super-family	TCDB subfamily
1.A.8	1.A.8.1 (1) 1.A.8.3 (1) 1.A.8.8 (6) 1.A.8.10 (1) 1.A.8.11 (1) 1.A.8.12 (1)
1.A.23	1.A.23.2 (1) 1.A.23.8 (1)
1.A.24	1.A.24.1 (3) 1.A.24.2 (1)
1.A.25	1.A.25.1 (4)
1.A.107	1.A.107.1 (2)
1.B.8	1.B.8.1 (1) 1.B.8.2 (1)
1.B.42	1.B.42.1 (3)
2.A.1	2.A.1.1 (1) 2.A.1.2 (11) 2.A.1.3 (1) 2.A.1.4 (1) 2.A.1.13 (2) 2.A.1.14 (3) 2.A.1.19 (3) 2.A.1.22 (1) 2.A.1.49 (1)
2.A.3	2.A.3.1 (2) 2.A.3.4 (3)
2.A.6	2.A.6.2 (1)
2.A.7	2.A.7.1 (2) 2.A.7.19 (1) 2.A.7.20 (1)
2.A.15	2.A.15.2 (1)
2.A.21	2.A.21.6 (1)
2.A.22	2.A.22.3 (1)
2.A.29	2.A.29.1 (2) 2.A.29.8 (1) 2.A.29.10 (1) 2.A.29.17 (1) 2.A.29.20 (1) 2.A.29.21 (1) 2.A.29.28 (1)
2.A.39	2.A.39.2 (2) 2.A.39.3 (1) 2.A.39.4 (1)
2.A.40	2.A.40.1 (1) 2.A.40.4 (1) 2.A.40.6 (1)
2.A.41	2.A.41.2 (1)
2.A.57	2.A.57.1 (1) 2.A.57.3 (1)
2.A.66	2.A.66.1 (4)
2.A.69	2.A.69.1 (3)
2.A.82	2.A.82.1 (3)
2.A.86	2.A.86.1 (1)
2.A.88	2.A.88.5 (1)
2.A.125	2.A.125.1 (1)
2.C.1	2.C.1.1 (1)
3.A.1	3.A.1.13 (1) 3.A.1.25 (3) 3.A.1.106 (1) 3.A.1.121 (1) 3.A.1.201 (4) 3.A.1.203 (5) 3.A.1.204 (6) 3.A.1.205 (1) 3.A.1.208 (3) 3.A.1.211 (5)

Table 35: Others mapping to TCDB families

For predicting the transported substrates, a summary of the experiments in Section 2 is presented in Table 36, ordered from the highest to lowest overall performance.

Feature	**MCC***
filtered-HMM-Profile	**0.64**
filtered-MSA-PAAC	0.60
filtered-MSA-AAC	0.56
MSA-PAAC	0.51
filtered-MSA-PseAAC	0.51
MSA-PseAAC	0.50
MSA-AAC	0.49
PSSM	0.39
PseAAC	0.27
PAAC	0.25
AAC	0.22
AAindex	0.01

Table 36: Overall cross-validation performance of different features

* MCC is calculated by using confusion matrix as in equation 17. Best cross-validation performance was achieved by filtered-HMM-Profile, filtered-MSA-PAAC also achieved high performance.

It is clear that there is a significant improvement when the evolutionary data is incorporated. For example, AAC without evolutionary data achieved an overall MCC of .22, and with the incorporation of evolutionary data MSA-AAC achieved MCC of .49. In addition, filtering the unreliable positions in the multiple sequence alignment further improved the overall performance. For example filtered-MSA-AAC overall MCC to .56.

This proves that the use homologous sequences in combination with filtering unreliable columns using TCS enhances the performance. The TM-COFFEE program, used for MSA, is tailored for sequences of membrane proteins, unlike other MSA algorithms that use general scoring schemes for soluble proteins. Specific conservation patterns were exposed by TM-COFFEE, resulting

in improvements with the alignment by aligning TMS with TMS. In addition, identifying unreliable positions in the multiple sequence alignment and eliminating them made the remaining alignment quite informative. To our knowledge, we are the first to combine MSA with PAAC. Also, we are the first to use filtered MSA with the protein compositions and HMMs.

Because HMM-profile effectively models the filtered MSA, filtered-HMM-profile achieved the best performance. Also, filtered-MSA-PAAC and filtered-MSA-AAC got comparable performance. Mishra *et al.s* [77] TrSSP method, is the latest published work in transporter substrate prediction which also claims to outperform other classifiers. Table 37 compares our results with those of Mishra *et al.* [77]

Class	**Specificity**		**Sensitivity**		**Accuracy**		**MCC**	
	TrSSP	X*	TrSSP	X	TrSSP	X	TrSSP	X
Amino acid	82.42	97.14	93.33	66.67	83.33	91.75	0.49	**0.67**
Anion	69.05	95.37	75.00	58.33	69.44	89.90	0.23	**0.53**
Cation	74.31	89.29	75.00	97.22	74.44	89.90	0.41	**0.80**
Electron	91.78	99.09	80.00	70.00	91.11	95.70	0.50	**0.76**
Protein	82.42	98.10	93.33	66.67	83.33	92.71	0.49	**0.70**
Sugar	76.79	97.22	91.67	75.00	77.78	93.68	0.38	**0.71**
Others	73.13	92.00	60.00	55.00	71.67	83.96	0.23	**0.47**
Overall					78.88	74.17	0.41	**0.68**

Table 37: Detailed substrate specificity performance

in Overall performance, We calculated accuracy as proportion of correct predictions divided by the total number of predictions, and MCC is calculated by usingh confusion matrix as in equation 17. while Mishra *et al.* calculated the overall accuracy and MCC as the average across the seven raws.

* X= filtered-HMM-Profiles

All substrate classes scored higher in accuracy and specificity. However, true positive rate (sensitivity) was higher in only one (cation) of the seven classes, while for the other six it was less than their results.Still, we were able to obtain an overall MCC rate of .68 in comparison

to .41 in Mishra *et al.*'s results. This finding indicates that our method is highly accurate on positive and negative classes with less misclassification of the two classes.

We can conclude from this project that using amino acid compositions alone does not yield strong prediction performance. The strengthened performance came from the incorporating evolutionary information while using specialized methods for transmembrane proteins. This finding suggest that the evolutionary information is key to classifying transmembrane proteins. Furthermore, certain positions in the alignment can have greater significance, so it is important to identify them. Using TCS score allowed us to eliminate some noise, but it only filtered about 50% of the sequences; further filtering is surely desirable. We strongly believe that there is still room for more experimentation and the improvement is possible.

Chapter V

Proposal for further research

The specific functions of many transmembrane proteins are still unknown due to the huge amount of available data and the exceptional challenges in the experimental characterization of their structure and function. Transporters are a major class in transmembrane proteins that move compounds across the membrane. The knowledge of the substrate specificities of a transporter is essential for understanding its physiological function. Also, transported substrates are important in the annotation of membrane proteins. Thus, computational approaches are needed to classify transmembrane proteins and predict their potential substrates. We aim to build a proteome-wide system that can determine the transporter substrate. This involves addressing the following research questions:

Q1: Given a protein sequence X, is this a transporter protein?

Q2: Given that protein Y is a transporter, what type of substrates does it transport across the membrane?

Here, I propose the main work items that we will wrok on in order to achieve our goal.

1 Distinguishing transporters from other types of proteins

This addresses the first research question —Given a protein sequence X, is this a transporter protein? It is important to apply substrate specificity detection methods (see Chapter III.2.2) after we ensure that the query protein is, in fact, a transporter. To address it, we need first to determine whether the queried protein is a transmembrane protein. In this regard, applying transmembrane topology prediction methods (see Chapter III. 1) to detect α-helical or β-barrel structures in the TMSs could be helpful. We need to take account of both efficiency and reliability in the used method since we are working at the proteome level rather than the

single sequence level. Nevertheless, detecting transmembrane proteins are not enough, as many receptors and enzymes have TMSs, but they are not transporters. So, computational methods are needed to distinguish between different transmembrane proteins classes. To our knowledge, no method was published yet to address this problem, as most use sequence similarity to detect the classes. Sequence similarity may give false classification because two proteins that belong to the same class do not always share significant sequence similarity [69]. We also seek to discover whether there is any correlation between the transmembrane protein topological orientations (N-terminus inside or outside of the membrane), the number of TMSs, and the class of the transmembrane protein.

2 Predicting Transporters substrate specificity

This is the second research question —Given that protein Y is a transporter, what type of substrates does it transport across the membrane? We addressed this question in our preliminary study (see Chapter IV), where we looked into the latest contribution in that area by Mishra *et al.* [77]. We used the same data with the same classes to evaluate our method. We achieved an overall MCC of .68 in comparison to a coefficient of .41 on their method. We believe that there is still room for more extermination and improvement. We desire to further improve our method by incorporating new techniques, such as profile to profile comparison, finding SDP, and modifying the AAindex algorithm. Furthermore, we saw that in substrate specificity prediction methods (see Chapter III.2.2) the prediction occurs at the level of substrate category or class (eg. amino acid) rather than the specific transported substrate (eg. arginine). While detecting the class can help experimentalists decrease the search space when determining the function of new protein sequences, the specific substrate is the optimal goal. The specific substrates can be documented using Chemical Entities of Biological Interest (ChEBI) ontology and we will aim to find the specific substrate once we improve the the class detection prediction.

One important point that all substrate specificity detection methods overlook is that the relationship between the transporter and the substrate is not one-to-one. For example, a transporter could transport more than one type of substrate. Granted, dealing with overlapping classes while single class methods are yet far from being established is illogical. We would look into the overlapping classes problem once we achieve an acceptable performance on the one class solutions.

3 Data collection and manual curation

As we discussed in Chapter I. 3, there is no gold standard database or dataset to work on in the context of transporters. Researchers tend to define their own datasets with a diverse number of substrate classes. For example, in Chapter III. 2.2, we saw that substrate-specific protein classes are not standardized. Some authors [71] choose to group the substrates into four groups, with one general class referring to all other types of substrate as others. Other authors (Schaadt *et al.* in [70] and [73]) decided to include oligopeptides —few amino acids linked in a polypeptide chain. Others (Chen *et al.* [71] and Mishra *et al.* [77]) elected to incorporate protein/mRNA, which consists of one or more polypeptides with at least 50 amino acids. Whereas others (Barghash *et al.* [74]) completely discounted the protein or oligopeptide category. Mishra *et al.* [77], who have the latest contribution in the substrate prediction method, have aspired to include the maximum possible number of classes according to their transported substrate (a total of seven classes with one categorized others). One question that needs to be addressed is whether having more classes is undoubtedly better, as in many cases, there is a possibility of an overlap between the different categories.

In 2000, Milton Saier [85], who established the TCDB, developed a system for the substrate specificity classification and cross-referenced it with the then-known TCDB families. His established classification system is shown in Table 38. Since there is an already established system, it is reasonable to move toward a standardized solution and start using it.

Category and substrate type	Subcategories
I.Inorganic molecules	A. Nonselective B. Water C. Cations D. Anions E. Others
II.Carbon compounds	A. Sugars, polyols, and their derivatives B. Monocarboxylates C. Di- and tricarboxylates D. Noncarboxylates organic anions (organophosphates, phosphonates, sulfonates, and sulfates) E. Others
III.Amino acids and their derivatives	A. Amino acids and conjugates B. Amines, amides, and polyamines C. Peptides D. Other related organocations E. Others
IV.Bases and their derivatives	A. (Nucleo)bases B. Nucleosides C. Nucleotides D. Other nucleobase derivatives E. Others
V.Vitamins, cofactors, and their precursors	A. Vitamins and vitamin or cofactor precursors B. Enzyme and redox cofactors C. Siderophores; siderophore-Fe complexes D. Signaling molecules E. Others
VI.Drugs, dyes, sterols, and toxics	A. Multiple drugs B. Specific drugs C. Bile salts and conjugates D. Sterols and conjugates
VII.Macromolecules	A. Carbohydrates B. Proteins C. Nucleic acids D. Lipids E. Others
VIII.Miscellaneous compounds	

Table 38: Classification of transport system substrates based on biological significance

We believe that applying the Saier's classification system will offer useful and practical subject for evaluation.

We need to build two datasets. The first dataset will be tailored to address the second research question. All of the proteins in this dataset will be transporter proteins. We plan to combine all the entries from TCDB and the entries that contain transport annotation in Swiss-Prot. The entries from Swiss-Prot will be manually checked to make sure that they are indeed transporters, as Swiss-Port annotations are not always consistent. Then, we manually need to categorize the transporters by Saier [85] classification.

The second dataset will be built to address the first research question —distinguishing transporters from other types of proteins. We need to include transmembrane proteins and non-transmembrane proteins.

We can get transmembrane proteins from UniProt by searching:

```
annotation:(type:transmem) AND reviewed:yes
```

For non transmembrane proteins:

```
NOT annotation:(type:transmem) AND reviewed:yes
```

Then we will intersect the first dataset with the retrieved transmembrane proteins data to categorize transmembrane proteins as transporters (the intersection), and annotate the other classes of transmembrane proteins.

4 Evaluating different substrate classification methods

Currently, a meaningful comparison of different substrate specificity classification methods (see Chapter III. 2.2) is impossible, as different methods use different subsets of substrate specificity classes. We plan to evaluate the performance of all the methods using the same dataset. This will be a useful resource to compare their results and move toward a standardized substrate classes.

5 Timeline

Here, I present a tentative timeline for this project, including the main milestones that we aspire to achieve during the course of my PhD.

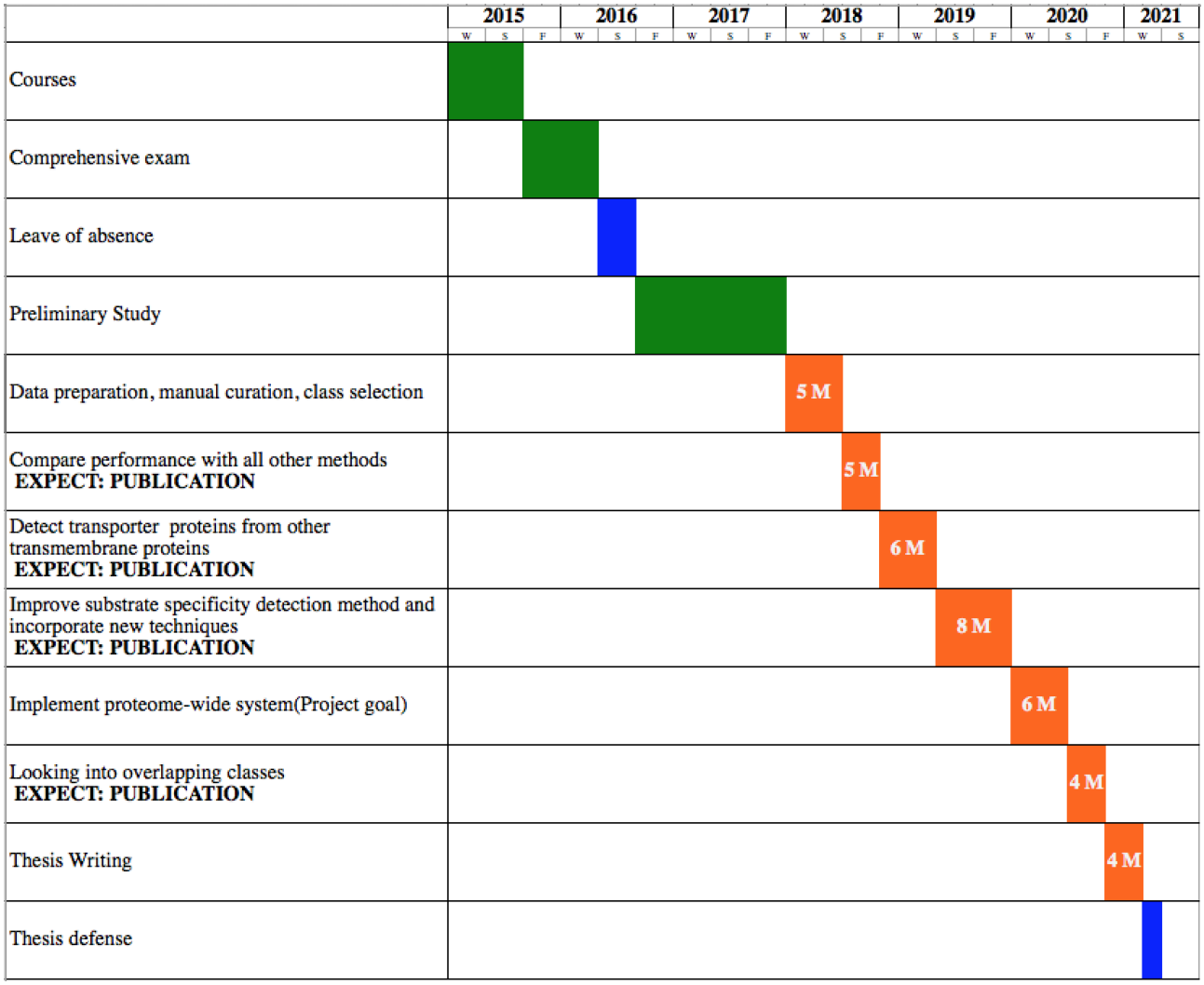

Figure 7: Tentative PhD timeline

Chapter VI

Conclusion

Transmembrane proteins play extremely important roles in all living cells; yet they are among the least-characterized proteins, owing to their unstable features. Transporters constitute a major class in transmembrane proteins that move the hydrophilic substrates across the hydrophilic membrane. Detecting the transmembrane transporter protein substrate specificity is beneficial in many levels such as annotation and drug design. Therefore, there is a pressing need to find computational solutions to predict the characterization of transmembrane proteins, which then can be subject to experimental validation. The main limitation that hinders such methods is the lack of available characterized proteins.

We to aim to built a proteome-wide system that can determine the transporter substrate. This involves distinguishing transmembrane protein, differentiating transporters from other types of transmembrane proteins and detecting the substrate specificity of the transporters. We made some progress in the area of substrate specificity detection in our preliminary study (see Chapter IV), where we confirmed that sequence similarity methods alone could give false information regarding the transported substrates. Hence, other features are needed to differentiate between different substrates. By filtering MSA we were able to achieve a better overall performance compared to the latest published work of Mishra *et al.* [77]. We plan to further improve our method and integrate new techniques, such as modifying the AAindex algorithm to combine amino acid compositions, orthology detection, and motif discovery for finding specificity-determining positions (SDPs). In addition, we plan to incorporate a larger datasets that is includes all of the annotated transports to-date. Moreover, we aspire to move toward standardized substrate classes, following Saiers classification system and predict the specific transported substrate rather than the general class.

Bibliography

[1] H. Lodish, A. Berk, L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, "Transport across cell membranes," in *Molecular Cell Biology*. New York: W. H. Freeman, 2000, ch. 15.

[2] L. Buehler, "The Structure of Membrane Proteins," in *Cell Membranes*. Garland Science, 2015, ch. 3.

[3] G. von Heijne, "Recent advances in the understanding of membrane protein assembly and structure," *Quarterly Reviews of Biophysics*, vol. 32, no. 4, pp. 285–307, 1999.

[4] G. E. Schulz, "Transmembrane β-barrel proteins," *Advances in Protein Chemistry*, vol. 63, pp. 47–70, 2003.

[5] D. Kozma, I. Simon, and G. E. Tusnády, "PDBTM: Protein data bank of transmembrane proteins after 8 years," *Nucleic acids Research*, vol. 41, no. D1, pp. D524–D529, 2013.

[6] M. M. Gromiha and Y.-Y. Ou, "Bioinformatics approaches for functional annotation of membrane proteins," *Briefings in Bioinformatics*, p. bbt015, 2013.

[7] D. Fotiadis, Y. Kanai, and M. Palacín, "The SLC3 and SLC7 families of amino acid transporters," *Molecular Aspects of Medicine*, vol. 34, no. 2, pp. 139–158, 2013.

[8] E. P. Carpenter, K. Beis, A. D. Cameron, and S. Iwata, "Overcoming the challenges of membrane protein crystallography," *Current Opinion in Structural Biology*, vol. 18, no. 5, pp. 581–586, 2008.

[9] A. B. Chang, R. Lin, W. K. Studley, C. V. Tran, and M. H. Saier, Jr, "Phylogeny as a guide to structure and function of membrane transport proteins (Review)," *Molecular Membrane Biology*, vol. 21, no. 3, pp. 171–181, 2004.

[10] H. Li, V. A. Benedito, M. K. Udvardi, and P. X. Zhao, "TransportTP: a two-phase classification approach for membrane transporter prediction and characterization," *BMC Bioinformatics*, vol. 10, no. 1, p. 418, 2009.

[11] Uniprot, "Uniprot," [Online; accessed Sep-2017]. [Online]. Available: http://www.uniprot.org/

[12] R. C. Edgar and S. Batzoglou, "Multiple Sequence Alignment," *Current Opinion in Structural Biology*, vol. 16, no. 3, pp. 368–373, 2006.

[13] J. Pevsner, "Multiple Sequence Alignment," in *Bioinformatics and Functional Genomics*, 2nd ed. Baltimore , Maryland: Wiley-Blackwell, 2009, ch. 6.

[14] K. Katoh and H. Toh, "Recent developments in the MAFFT multiple sequence alignment program," *Briefings in Bioinformatics*, vol. 9, no. 4, pp. 286–298, 2008.

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

[16] J. Daugelaite, A. O'Driscoll, and R. D. Sleator, "An overview of multiple sequence alignments and cloud computing in bioinformatics," *ISRN Biomathematics*, vol. 2013, 2013.

[17] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding *et al.*, "Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega," *Molecular Systems Biology*, vol. 7:539, no. 1, 2011.

[18] G. Blackshields, F. Sievers, W. Shi, A. Wilm, and D. G. Higgins, "Research sequence embedding for fast construction of guide trees for multiple sequence alignment," *Algorithms for Molecular Biology*, vol. 5:21, 2010.

[19] J. Söding, "Protein homology detection by HMM–HMM comparison," *Bioinformatics*, vol. 21, no. 7, pp. 951–960, 2005.

[20] K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform," *Nucleic Acids Research*, vol. 30, no. 14, pp. 3059–3066, 2002.

[21] C. Notredame, D. G. Higgins, and J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment," *Journal of Molecular Biology*, vol. 302, no. 1, pp. 205–217, 2000.

[22] W. Pirovano, K. A. Feenstra, and J. Heringa, "PRALINE: a strategy for improved multiple alignment of transmembrane proteins," *Bioinformatics*, vol. 24, no. 4, pp. 492–497, 2008.

[23] J.-M. Chang, P. Di Tommaso, J.-F. Taly, and C. Notredame, "Accurate multiple sequence alignment of transmembrane proteins with PSI-Coffee," *BMC Bioinformatics*, vol. 13, no. Suppl 4, p. S1, 2012.

[24] Y. Shafrir and H. R. Guy, "STAM: simple transmembrane alignment method," *Bioinformatics*, vol. 20, no. 5, pp. 758–769, 2004.

[25] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[26] Y. Liu, B. Schmidt, and D. L. Maskell, "MSAProbs: multiple sequence alignment based on pair hidden markov models and partition function posterior probabilities," *Bioinformatics*, vol. 26, no. 16, pp. 1958–1964, 2010.

[27] E. Krissinel, "On the relationship between sequence and structure similarities in proteomics," *Bioinformatics*, vol. 23, no. 6, pp. 717–723, 2007.

[28] O. Gotoh, "Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments," *Journal of Molecular Biology*, vol. 264, no. 4, pp. 823–838, 1996.

[29] J. D. Thompson, B. Linard, O. Lecompte, and O. Poch, "A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives," *PloS one*, vol. 6, no. 3, p. e18093, 2011.

[30] F. S.-M. Pais, P. de Ruy, G. Oliveira, and R. Coimbra, "Assessing the efficiency of multiple sequence alignment programs." *Algorithms for Molecular Biology*, vol. 9, no. 4, 2014.

[31] K. Nishikawa, Y. Kubota, and O. Tatsuo, "Classification of proteins into groups based on amino acid composition and other characters. I. Angular distribution," *Journal of Biochemistry*, vol. 94, no. 3, pp. 981–995, 1983.

[32] H. Nakashima, K. Nishikawa, and O. Tatsuo, "The folding type of a protein is relevant to the amino acid composition," *Journal of Biochemistry*, vol. 99, no. 1, pp. 153–162, 1986.

[33] G. E. Tusnady and I. Simon, "Principles governing amino acid composition of integral membrane proteins: application to topology prediction," *Journal of Molecular Biology*, vol. 283, no. 2, pp. 489–506, 1998.

[34] K.-C. Chou and C.-T. Zhang, "Prediction of protein structural classes," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 30, no. 4, pp. 275–349, 1995.

[35] J. Cedano, P. Aloy, J. A. Perez-Pons, and E. Querol, "Relation between amino acid composition and cellular location of proteins," *Journal of Molecular Biology*, vol. 266, no. 3, pp. 594–600, 1997.

[36] M. M. Gromiha and Y. Yabuki, "Functional discrimination of membrane proteins using machine learning techniques," *BMC Bioinformatics*, vol. 9, no. 1, p. 135, 2008.

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

[38] C. Tanford, "Contribution of hydrophobic interactions to the stability of the globular conformation of proteins," *Journal of the American Chemical Society*, vol. 84, no. 22, pp. 4240–4247, 1962.

[39] T. P. Hopp and K. R. Woods, "Prediction of protein antigenic determinants from amino acid sequences," *Proceedings of the National Academy of Sciences*, vol. 78, no. 6, pp. 3824–3828, 1981.

[40] J. Kyte and R. F. Doolittle, "A simple method for displaying the hydropathic character of a protein," *Journal of Molecular Biology*, vol. 157, no. 1, pp. 105–132, 1982.

[41] G. Von Heijne, "Membrane hydrophobicity protein structure prediction analysis and the positive-inside," *Journal of Molecular Biology*, vol. 225, pp. 487–94, 1992.

[42] T. Nugent and D. T. Jones, "Transmembrane protein topology prediction using support vector machines," *BMC Bioinformatics*, vol. 10:159, no. 1, 2009.

[43] K. D. Tsirigos, A. Hennerdal, L. Käll, and A. Elofsson, "A guideline to proteome-wide α-helical membrane protein topology predictions," *Proteomics*, vol. 12, no. 14, pp. 2282–2294, 2012.

[44] K. D. Tsirigos, C. Peters, N. Shu, L. Käll, and A. Elofsson, "The TOPCONS web server for consensus prediction of membrane protein topology and signal peptides," *Nucleic Acids Research*, vol. 43, no. W1, pp. W401–W407, 2015.

[45] A. Bernsel, H. Viklund, A. Hennerdal, and A. Elofsson, "TOPCONS: consensus prediction of membrane protein topology," *Nucleic Acids Research*, vol. 37, no. suppl_2, pp. W465–W468, 2009.

[46] S. M. Reynolds, L. Käll, M. E. Riffle, J. A. Bilmes, and W. S. Noble, "Transmembrane topology and signal peptide prediction using dynamic bayesian networks," *PLoS Computational Biology*, vol. 4, no. 11, p. e1000213, 2008.

[47] L. Käll, A. Krogh, and E. L. Sonnhammer, "An HMM posterior decoder for sequence feature prediction that includes homology information," *Bioinformatics*, vol. 21, no. suppl_1, pp. i251–i257, 2005.

[48] H. Viklund, A. Bernsel, M. Skwark, and A. Elofsson, "SPOCTOPUS: a combined predictor of signal peptides and membrane protein topology," *Bioinformatics*, vol. 24, no. 24, pp. 2928–2929, 2008.

[49] H. Viklund and A. Elofsson, "OCTOPUS: improving topology prediction by two-track ANN-based preference scores and an extended topological grammar," *Bioinformatics*, vol. 24, no. 15, pp. 1662–1668, 2008.

[50] A. Bernsel, H. Viklund, J. Falk, E. Lindahl, G. von Heijne, and A. Elofsson, "Prediction of membrane-protein topology from first principles," *Proceedings of the National Academy of Sciences*, vol. 105, no. 20, pp. 7177–7181, 2008.

[51] F. S. Berven, K. Flikka, H. B. Jensen, and I. Eidhammer, "BOMP: a program to predict integral β-barrel outer membrane proteins encoded within genomes of Gram-negative bacteria," *Nucleic Acids Research*, vol. 32, pp. W394–W399, 2004.

[52] J. Hu and C. Yan, "A method for discovering transmembrane β-barrel proteins in gram-negative bacterial proteomes," *Computational Biology and Chemistry*, vol. 32, no. 4, pp. 298–301, 2008.

[53] I. Jacoboni, P. L. Martelli, P. Fariselli, V. De Pinto, and R. Casadio, "Prediction of the transmembrane regions of β-barrel membrane proteins with a neural network-based predictor," *Protein Science*, vol. 10, no. 4, pp. 779–787, 2001.

[54] Y. Ou, M. M. Gromiha, S. Chen, and M. Suwa, "TMBETADISC-RBF: discrimination of β-barrel membrane proteins using RBF networks and PSSM profiles," *Computational Biology and Chemistry*, vol. 32, no. 3, pp. 227–231, 2008.

[55] P. G. Bagos, T. D. Liakopoulos, I. C. Spyropoulos, and S. J. Hamodrakas, "PRED-TMBB: a web server for predicting the topology of β-barrel outer membrane proteins," *Nucleic acids research*, vol. 32, no. suppl_2, pp. W400–W404, 2004.

[56] N. K. Singh, A. Goodman, P. Walter, V. Helms, and S. Hayat, "TMBHMM: a frequency profile based HMM for predicting the topology of transmembrane beta barrel proteins and the exposure status of transmembrane residues," *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, vol. 1814, no. 5, pp. 664–670, 2011.

[57] S. Hayat and A. Elofsson, "BOCTOPUS: improved topology prediction of transmembrane β barrel proteins," *Bioinformatics*, vol. 28, no. 4, pp. 516–522, 2012.

[58] K. D. Tsirigos, A. Elofsson, and P. G. Bagos, "PRED-TMBB2: improved topology prediction and detection of beta-barrel outer membrane proteins," *Bioinformatics*, vol. 32, no. 17, pp. i665–i671, 2016.

[59] Y.-y. Ou, S.-a. Chen, and M. M. Gromiha, "Prediction of membrane spanning segments and topology in β-barrel membrane proteins at better accuracy," *Journal of Computational Chemistry*, vol. 31, no. 1, pp. 217–223, 2010.

[60] A. G. Garrow, A. Agnew, and D. R. Westhead, "TMB-Hunt: an amino acid composition based method to screen proteomes for beta-barrel transmembrane proteins," *BMC Bioinformatics*, vol. 6, no. 1, p. 56, 2005.

[61] H. Lin, "The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 252, no. 2, pp. 350–356, 2008.

[62] P. G. Bagos, T. D. Liakopoulos, and S. J. Hamodrakas, "Evaluation of methods for predicting the topology of β-barrel outer membrane proteins and a consensus prediction method," *BMC Bioinformatics*, vol. 6, no. 1, p. 7, 2005.

[63] M. Remmert, D. Linke, A. N. Lupas, and J. Söding, "HHomppredicitn and classification of outer membrane proteins," *Nucleic Acids Research*, vol. 37, no. suppl_2, pp. W446–W451, 2009.

[64] G. E. Tusnady and I. Simon, "The HMMTOP transmembrane topology prediction server," *Bioinformatics*, vol. 17, no. 9, pp. 849–850, 2001.

[65] E. L. Sonnhammer, G. Von Heijne, A. Krogh *et al.*, "A hidden markov model for predicting transmembrane helices in protein sequences," in *Ismb*, vol. 6, 1998, pp. 175–182.

[66] L. Käll and E. L. Sonnhammer, "Reliability of transmembrane predictions in whole-genome data," *FEBS letters*, vol. 532, no. 3, pp. 415–418, 2002.

[67] F. Aplop and G. Butler, "On predicting transport proteins and their substrates for the reconstruction of metabolic networks," in *Computational Intelligence in Bioinformatics and Computational Biology*. IEEE, 2015, pp. 1–9.

[68] M. H. Saier, "Molecular phylogeny as a basis for the classification of transport proteins from bacteria, archaea and eukarya," *Advances in Microbial Physiology*, vol. 40, pp. 81–136, 1998.

[69] J. C. Whisstock and A. M. Lesk, "Prediction of protein function from protein sequence and structure," *Quarterly Reviews of Biophysics*, vol. 36, no. 03, pp. 307–340, 2003.

[70] N. S. Schaadt, J. Christoph, and V. Helms, "Classifying substrate specificities of membrane transporters from Arabidopsis Thaliana," *Journal of Chemical Information and Modeling*, vol. 50, no. 10, pp. 1899–1905, 2010.

[71] S. Chen, Y. Ou, T. Lee, and M. M. Gromiha, "Prediction of transporter targets using efficient RBF networks with PSSM profiles and biochemical properties," *Bioinformatics*, vol. 27, no. 15, pp. 2062–2067, 2011.

[72] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "AAindex: amino acid index database, progress report 2008," *Nucleic Acids Research*, vol. 36, pp. D202–D205, 2007.

[73] N. Schaadt and V. Helms, "Functional classification of membrane transporters and channels based on filtered TM/non-TM amino acid composition," *Biopolymers*, vol. 97, no. 7, pp. 558–567, 2012.

[74] A. Barghash and V. Helms, "Transferring functional annotations of membrane transporters on the basis of sequence similarity and sequence motifs," *BMC Bioinformatics*, vol. 14, no. 1, p. 343, 2013.

[75] R. D. Finn, J. Clements, and S. R. Eddy, "HMMER web server: interactive sequence similarity searching," *Nucleic Acids Research*, vol. 39, no. suppl_2, pp. W29–W37, 2011.

[76] T. L. Bailey, C. Elkan *et al.*, "Fitting a mixture model by expectation maximization to discover motifs in bipolymers," 1994.

[77] N. K. Mishra, J. Chang, and P. X. Zhao, "Prediction of membrane transport proteins and their substrate specificities using primary sequence information," *PLoS One*, vol. 9, no. 6, p. e100278, 2014.

[78] S. Kawashima and M. Kanehisa, "AAindex: amino acid index database," *Nucleic Acids Research*, vol. 28, no. 1, pp. 374–374, 2000.

[79] J.-M. Chang, P. Di Tommaso, and C. Notredame, "TCS: a new multiple sequence alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction," *Molecular Biology and Evolution*, pp. 1625–1637, 2014.

[80] S. R. Eddy, "HMMER: Profile hidden Markov models for biological sequence analysis," 2001.

[81] Z. Ding, "Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics," 2011.

[82] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[83] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation measures for models assessment over imbalanced data sets," *Journal Of Information Engineering and Applications*, vol. 3, no. 10, 2013.

[84] J. Gorodkin, "Comparing two K-category assignments by a K-category correlation coefficient," *Computational biology and chemistry*, vol. 28, no. 5, pp. 367–374, 2004.

[85] M. H. Saier, "A functional-phylogenetic classification system for transmembrane solute transporters," *Microbiology and Molecular Biology Reviews*, vol. 64, no. 2, pp. 354–411, 2000.

[86] A. B. Chang, R. Lin, W. K. Studley, C. V. Tran, and M. H. Saier, Jr, "Phylogeny as a guide to structure and function of membrane transport proteins," *Molecular Membrane Biology*, vol. 21, no. 3, pp. 171–181, 2004.

[87] A. Reddy, J. Cho, S. Ling, V. Reddy, M. Shlykov, and M. H. Saier, "Reliability of nine programs of topological predictions and their application to integral membrane channel and carrier proteins," *Journal of Molecular Microbiology and Biotechnology*, vol. 24, no. 3, pp. 161–190, 2014.

[88] A. Chang, M. Scheer, A. Grote, I. Schomburg, and D. Schomburg, "BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009," *Nucleic acids research*, vol. 37, no. suppl_1, pp. D588–D592, 2008.

[89] D. Vodopich and R. Moore, *Biology Laboratory Manual*, 6th ed., Minneapolis.

[90] Wikipedia, "Cell membrane," 2017, [Online; accessed Sep-2017]. [Online]. Available: https://en.wikipedia.org/wiki/Cell_membrane

[91] P. Raven and G. Johnson, "Membranes," in *Biology*, 6th ed. Boston, MA: McGraw-Hill Higher Education, 2008, ch. 6.

[92] P. Baldi and S. Brunak, "Proteins and Proteomics," in *Bioinformatics: the Machine Learning Approach*, 2nd ed. MIT press.

[93] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2014.

[94] D. Mount, "Phylogenetic Prediction," in *Bioinformatics: Sequence and Genome Analysis*, 2nd ed. New York: Cold Spring Harbor Lab Press, 2004, ch. 7.

[95] C. S. Clair and J. Visick, "Tree Building in Molecular Phylogenetics: Tree Domains of Life," in *Exploring Bioinformatics: A Project-Based Approach*, 2nd ed. Burlington: Jones & Bartlett Learning, 2013, ch. 7.

[96] J. Pevsner, "Molecular Phylogeny and Evolution," in *Bioinformatics and Functional Genomics*, 2nd ed. Baltimore , Maryland: Wiley-Blackwell, 2009, ch. 7.

[97] R. Sokal and C. Michener, "A statistical method for evaluating systematic relationships," *University of Kansas Scientific Bulletin*, vol. 27, pp. 409–1438, 1958.

[98] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees." *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406–425, 1987.

[99] R. Eck and M. O. Dayhoff, *Atlas of Protein Sequence and Structure.* Minneapolis: Silver Spring.

[100] J. Felsenstein, "Evolutionary trees from DNA sequences: a maximum likelihood approach," *Journal of molecular evolution*, vol. 17, no. 6, pp. 368–376, 1981.

[101] Z. Yang, "Phylogeny Reconstruction: Overview," in *Molecular Evolution: A Statistical Approach.* OUP Oxford, 2008, p. 81.

[102] J. Felsenstein, "Cases in which parsimony or compatibility methods will be positively misleading," *Systematic Biology*, vol. 27, no. 4, pp. 401–410, 1978.

[103] K. Strimmer and A. Von Haeseler, "Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies," *Molecular Biology and Evolution*, vol. 13, no. 7, pp. 964–969, 1996.

[104] C. Struck, "Amino acid uptake in rust fungi," *Frontiers in Plant Science*, vol. 6, 2015.

[105] A. Pavlopoulou and I. Michalopoulos, "State-of-the-art bioinformatics protein structure prediction tools (review)," *International Journal of Molecular Medicine*, vol. 28, no. 3, pp. 295–310, 2011.

[106] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan *et al.*, "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003," *Nucleic Acids Research*, vol. 31, no. 1, pp. 365–370, 2003.

[107] K. Tamura, G. Stecher, D. Peterson, A. Filipski, and S. Kumar, "MEGA6: molecular evolutionary genetics analysis version 6.0," *Molecular Biology and Evolution*, vol. 30, no. 12, pp. 2725–2729, 2013.

[108] T. Muth, J. A. García-Martín, A. Rausell, D. Juan, A. Valencia, and F. Pazos, "JDET: interactive calculation and visualization of function-related conservation patterns in multiple sequence alignments and structures," *Bioinformatics*, vol. 28, no. 4, pp. 584–586, 2012.

[109] NCBI, "NCBI ftp site," [Online; accessed Sep-2017]. [Online]. Available: ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nr.gz