ON PREDICTING TRANSMEMBRANE PROTEINS

By
Munira Alballa

SUBMITTED AS COMPREHESIVE EXAM REPROT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
AT
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC
JANUARY 2016

Contents

List of Figures

List of Tables

Acronyms

AAAP Amino Acid/Auxin Permease

AAC Amino Acid Composition

ACT Amino Acid/Choline Transporter

APC Amino Acid-Polyamine-Organocation

ATP Adenosine Triphosphate

BLAST Basic Local Alignment Search Tool

CEMC Cross Entropy Monte Carlo

HMM Hidden Markov Model

IMP Integral Membrane Proteins

IUBMB International Union of Biochemistry and Molecular Biology

KNN K-Nearest Neighbors

LAT L-type Amino Acid Transporter

LOOCV Leave-one-out cross-validation

MCMC Markov Chain Monte Carlo

MEGA Molecular Evolutionary Genetics Analysis

ML Maximum Likelihood

MSA Multiple Sequence Alignment

NJ Neighbor Joining

PAAC Pair Amino Aside composition

PDB Protein Data Bank

PseAAC Pseudo-Amino Acid Composition

SDP Specificity-Determining Positions

T-COFFEE Tree based Consistency Objective Function for Alignment Evaluation

TCDB Transporter Classification Database

TCID Transport Classification Identification

TMS Transmembrane Segments

TSC Transitive Consistency Score

YAT Yeast Amino Acid Transporter

Glossary

Accuracy The ability of the classifier to find the number of correct decision (both positive and negative) among the total number of cases examined.

ATP hydrolysis The reaction by which chemical energy that has been stored in the high-energy phosphoanhydride bonds in adenosine triphosphate (ATP) is release.

Bayesian method A character-based method for the creation of phylogenetic trees that uses Bayesian statistics. Available in: `http://mrbayes.sourceforge.net/`

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.

Clustal Omega [22] The latest multiple sequence alignment program from the Clustal family. Can be downloaded from: `http://www.clustal.org/omega/` Can be run online from the EBI web server: `http://www.ebi.ac.uk/Tools/msa/clustalo/`

ClustalW [20] A commonly used progressive multiple sequence alignment program. Can be downloaded from: `http://www.clustal.org/clustal2/` Can be run online from the EBI web server: `http://www.ebi.ac.uk/Tools/msa/clustalw2/`

Concentration gradient The process of particles moving through a solution from an area of higher number of particles to an area of lower number of particles.

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

figTree [62] figTree is designed as a graphical viewer of phylogenetic trees and as a program for producing publication-ready figures. Can be downloaded from: `http://tree.bio.ed.ac.uk/software/figtree/`

HMMTOP [57] A topology prediction method to find the number of transmembrane helices in integral membrane proteins. Can be downloaded from: `http://www.enzim.hu/hmmtop/html/download.html` Can be run online from: `http://www.enzim.hu/hmmtop/html/adv_submit.html`

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.

MAFFT [25] A highly efficient iterative multiple sequence alignment program. Can be downloaded from: `http://mafft.cbrc.jp/alignment/software/` Can be run online from: `http://mafft.cbrc.jp/alignment/server/`

Maximum likelihood [42] A character-based method for the creation of phylogenetic trees. Available in many software tools for phylogeny, such as MEGA: `http://www.megasoftware.net/`

Maximum parsimony [41] A character-based method for the creation of phylogenetic trees. Maximum parsimony assumes that the rate of amino acid substitution is constant for all the branches in the tree. Available in many software tools for phylogeny, such as MEGA: `http://www.megasoftware.net/`

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

NJ [40] A hierarchical clustering method for the creation of phylogenetic trees. Available in many software tools for phylogeny, such as MEGA: `http://www.megasoftware.net/`

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.

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.

Sensitivity The ability of the classifier to detect the positives that are correctly identified as such.

SIWSS-PROT [58] A high quality annotated and non-redundant protein sequence database. http://www.uniprot.org/uniprot/?query=*&fil=reviewed%3Ayes

Specificity The ability of the classifier to detect the negatives that are correctly identified as such.

T-COFFEE [26] A consistency-based multiple sequence alignment method. It is considered one of the most accurate available programs based on benchmarking studies. Can be downloaded from: http://www.tcoffee.org/Projects/tcoffee/index.html#DOWNLOAD Can be run online from: http://tcoffee.crg.cat/apps/tcoffee/do:regular

TCS [61] 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 TCS evaluation and filtering procedure is implemented in the T-Coffee package and can be used to evaluate and filter any third party multiple sequence alignment. Can be downloaded with T-COFFEE package: http://www.tcoffee.org/Projects/tcoffee/index.html#DOWNLOAD Can be run online from: http://tcoffee.crg.cat/apps/tcoffee/do:core

TM-COFFEE [28] TM-COFFEE is part of the T-COFFEE package that is designed specifically to align transmembrane proteins. Can be downloaded from: http://www.tcoffee.org/Projects/tcoffee/index.html#DOWNLOAD Can be run online from: http://tcoffee.crg.cat/apps/tcoffee/do:tmcoffee

TMHMM[56] A topology prediction method to find the number of transmembrane helices in integral membrane proteins. Can be run online from: http://www.cbs.dtu.dk/services/TMHMM/

UPGMA [39] A hierarchical clustering method for the creation of phylogenetic trees. UPGMA assumes that the rate of amino acid substitution is constant for all the branches in the tree. Available in many software tools for phylogeny, such as MEGA: http://www.megasoftware.net/

Abstract

Transmembrane proteins are essential in all living cells. They enable vital cell functions-such as regulation, metabolism and energy production. It is estimated that more than 25% of an organisms complete genome is made up of transmembrane proteins. Any slight dysfunction of transmembrane proteins can cause fatal diseases. For this reason, they recently became an attractive target for many pharmaceutical companies. Experimental characterization of their structure and function is exceptionally difficult owing to their hydrophobic surfaces and their lack of stability; making them one of the least characterized proteins. Therefore, there is an urgent need for computational approaches that are able to distinguish transmembrane proteins and predict their function. Many initial attempts were made to classify and differentiate transmembrane proteins. Yet, this area of research is still in early stages and we are far behind finding a global solution. In this report we discuss the state-of-the-art techniques and highlight their potentials and limitations. In addition, we apply different techniques and compare their performance. In the end, we conclude with what need to be done and the future direction and limitations.

1 Introduction

1.1 Biological background

The fact that cell membranes are the only cellular structure that is found in all cells of all organisms on earth underlines their biological importance [2]. Not only does the cell membrane maintain the integrity of the cell by separating the critical chemicals and structures needed to maintain the cell from the surrounding environment, but also it acts as a selectively permeable barrier between the cell and the extracellular environment [3]. The selective permeability regulates the movement of molecules across the membrane so that the essential molecules such as sugar, amino acids, phosphates, and lipids enter the cell while waste compounds leave the cell.

Figure 1 illustrates the main components of the cell membrane. The basic structure of all cellular membranes is the phospholipid bilayer that consists of two layers of phospholipid molecules whose fatty tails forms the hydrophobic interior of the bilayer, and their hydrophilic polar heads line both inside and outside the cell surface. The phospholipid bilayer is embedded with membrane proteins that can be either integral or peripheral.

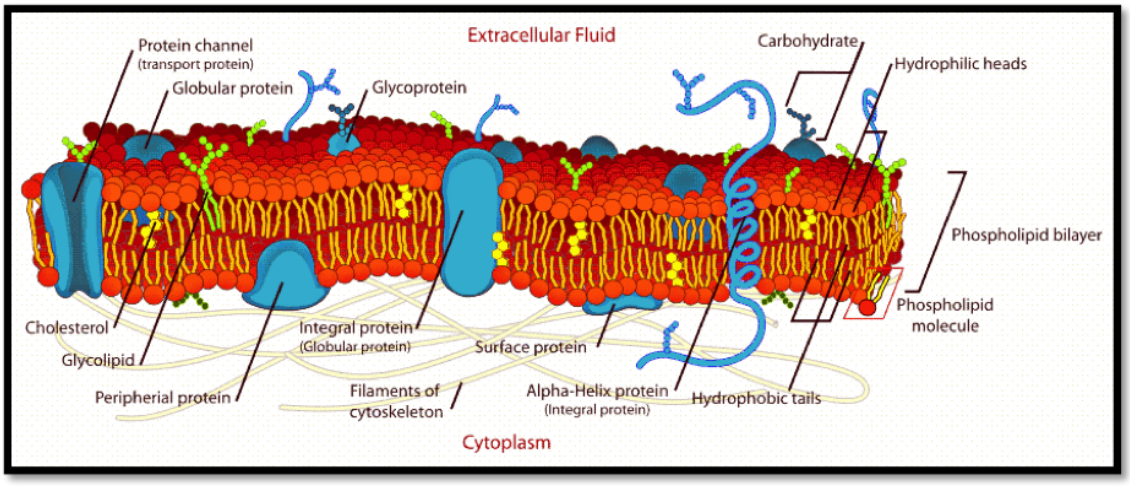

Figure 1: The structure of the Cell membrane [1]

The membrane proteins with which we are concerned in this report are the integral membrane proteins (IMP). The integral membrane proteins are permanently

attached to the biological membrane and can have one or more transmembrane segments (TMS) embedded in the phospholipid bilayer.

IMPs are divided into two groups: the first and most common is the integral polytopic protein that spans the entire phospholipid bilayer. This type is mostly referred to as the transmembrane protein. The second type is the integral monotopic protein, which is associated to the membrane from one side only. The focus of this report is on the first type, which will be referred to as the transmembrane protein from this point forward.

1.1.1 Transmembrane Proteins

The majority of the molecules that enter and leave cell membranes do so with the help of transmembrane proteins. Transmembrane transport proteins can be classified based on their function into three main classes: pumps, channels, and transporters (or carriers). Pumps use ATP hydrolysis energy to move ions or small molecules across a membrane against a chemical concentration gradient or electrical potential. Channel proteins transport water or ions down their concentration without the need of energy; they form a passageway that allows multiple water molecules or ions to move simultaneously in either directions across membranes. Transporters transport ions as well as other solutes like sugar and amino acids across the membrane. Unlike channel proteins, they facilitate the movement of ions and molecules across the membrane by physically binding to one or a few substrate molecules at a time on one side of the membrane, and releasing them on the other [2].

Three types of transporters have been identified as shown in Figure 2. Uniporters bind to one molecule at a time and transport it with its concentration gradient. Antiporters move two molecules in opposite directions, one molecule against its concentration and the other with its concentration gradient. Symporters transport two molecules in the same direction, like antiporters, one molecule against its concentration and the other with its concentration gradient. Because symporters and antiporters move certain molecules against their concentration gradient, they are often called active transporters. However, they do not directly hydrolyze ATP during the transport [4].

The secondary structure of transmembrane proteins can be either α-helix or β-

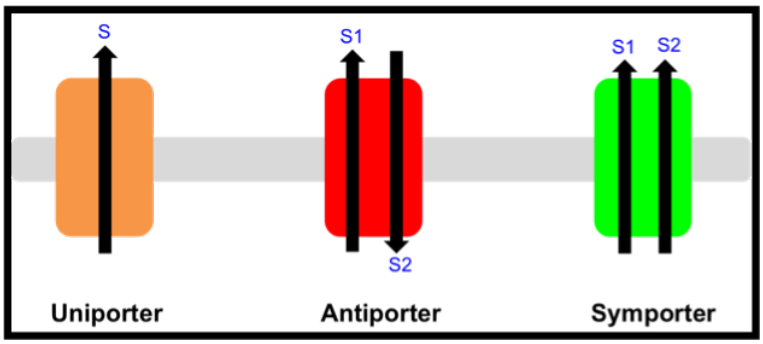

Figure 2: The three groups of transporters

barrel. The commonly seen α-helical transmembrane proteins are located in the inner membranes of bacterial cells and in the plasma membranes of eukaryotes. Their membrane spanning segments are formed by the connection of α-helices with hydrophilic loops. The β-barrel transmembrane proteins are only found in outer membrane of Gram-negative bacteria. Their membrane-spanning segments are antiparallel β-strands that form a barrel-like channel.

1.2 Challenges and Motivations

Transmembrane proteins play several important roles in the living cells, such as regulation, metabolism, and energy production. More than a quarter of protein sequences in genomes are identified as transmembrane proteins [5]. Any malfunction of these proteins can disturb the body homoeostasis giving a raise to many human diseases [6]. For this reason, transmembrane proteins have become very attractive targets for the pharmaceutical industry; over half of todays drugs have some effect on them [7]. Although 25% of the protein sequences in genomes are identified as transmembrane proteins, they are still not very well characterized. In general, predicting protein function experimentally is not an easy task, because the function may be related specifically to the native environment in which a particular organism lives [8]. In addition, membrane proteins are especially difficult to study for many reasons. For example, they have 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 [9].

For example, as of October 2015, Protein Data Bank (PDB) contains more than

112,000 Protein structures, and only 1% of those represent the membrane proteins. Consequently, the classification of transporters according to their substrate specificity along with their families or subfamilies remains a challenge toward the advancement of both structural and functional biology.

From here comes the need for advanced computational techniques that use the available experimental data to predict the membrane transporter proteins and their functions. The next section will highlight some of the developed techniques.

1.3 Transmembrane Protein Classification

According to Aplop *et al.* [10] the efforts that were made could either classify a transporter based on the family to which it belongs to or according to the substrate it transports (e.g., amino acids, hexose). The classification into families commonly follows Transporter Classification Database (TCDB) [11]. TCDB uses the International Union of Biochemistry and Molecular Biology (IUBMB) approved classification system. TCDB is a curated database of accurate and experimentally characterized information from over 10,000 published references. As of October 2015, 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. It is worth mentioning that the same substrate may belong to different families and a single family may transport many different substrates.

Many of the earlier bioinformatics efforts classified transporter proteins to their corresponding putative families by using multiple sequence alignments and phylogeny [12] [13]. The rationale behind using those techniques is that proteins of 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 [13]. The main limitation of these approaches, however, is that homologous

sequences do not always share significant sequence similarity. Similarly, proteins with high sequence similarity do not always share the same function [14]. In the same manner, it is often impossible to predict the transported substrate based on these methods, because two proteins that transport the same substrate may belong to a different superfamily. For example, two protein transporters that transport aromatic amino acids belong to distinctly related superfamilies (2.A.42.1.5 and 2.A.3.1.12) with only 18.4% sequence identity.

Other methods incorporate machine-learning techniques to predict the class of a transporter. For example Gromiha *et al.* [15] found that neural networks achieved the highest accuracy compared to other machine-learning algorithms when using amino acid composition features to classify transporters as channels/pores, electrochemical and active transporters. Another example is that TransportTP [11] classifies a transporter to TCDB families in two phases; the first phase uses 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 improve the initial prediction by collecting different features using non-homology and homology evidence from other sources. None of these methods, however, detect the substrate specificity of the query protein.

The number of studies that predict the substrate specificity of a transport protein is quite limited. The paper with the highest published accuracy by Helms *et al.* [16] used amino acid composition, higher sequence order information, amino acid characteristics, and sequence conservation to measure the similarity of membrane transporters from Arabidopsis thaliana. Four substrate classes were considered in the classification, namely, amino acids, oligopeptides, phosphates, and hexoses. This paper shows that integrating additional information to the commonly used amino acid frequency leads to an improved prediction performance to 90% or more.

In this report we illustrate the advantages and limitations of using phylogeny to classify a transmembrane protein. In addition, we implement bioinformatics techniques by Helms *et al.* [16] and discuss their potentials and limitations. The rest of this report is organized as follows: section 2 gives a brief background of some bioinformatics tools and techniques that are used to classify transmembrane proteins. Section 3 outlines the project that was developed as part of this comprehensive exam and discusses the results. Finally, section 4 concludes this report.

2 Bioinformatics Methods

2.1 Multiple Sequence Alignment

Multiple sequence alignments (MSA) are fundamental tools for protein structure, function prediction, phylogenetic analysis, and other bioinformatics and molecular evolutionary applications. 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 [17] [18]. MSA development is an 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 [19].

ClustalW [20], one of the most popular MSA heuristic algorithms, uses 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 [18] [21].

Clustal Omega [22], the latest algorithm from the Clustal family, is highly efficient

and more sensitive 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 [23], 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 [24] 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 [25] 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 [26], 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 because two proteins may share only 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 [27].

Few packages have been published to tackle the problem of aligning transmembrane proteins, such as PROLIN-TM [27], TM-COFFEE [28] and STAM [29].

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.* [28]. The TM-COFFEE algorithm can be summarized as follows: for each sequence in need of alignmnet, perform a homology search using BLAST [30] 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 10% improvement to the MSAProbs [31], 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 [17].

The assessment of MSA has been subject for research in the last few 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 do this is by comparing 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 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 [32] [33].

Several benchmark datasets have been created to be 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 [17] [34] [35]. 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 [35]. Table 1 summarizes the advantages and disadvantages and gives general recommendations based on the recommendation of the comparative benchmarking studies.

2.2 Phylogenetic Trees

A phylogenetic analysis of related protein sequences is done to determine how the sequences might have derived during evolution. A phylogenetic tree is a diagram that contains branches and nodes. Leafs of the tree have the available nucleic acid or protein sequences that we are analyzing. The branching relationships of the inner part of the tree reflect the degree to which the sequences are different. Closely related sequences are located in neighboring branches that are joined together, while less related sequences are on branches that are more distant from each other [36]. A phylogenetic tree provides a guide to function and structure. Two related proteins that have a common ancestor are expected to have similar structures and function in proportion to the their sequence similarity, but two independently evolving proteins should not [13].

The first step to building a tree is to perform MSA. Hence, the quality of the algorithm used plays an important role in producing a biologically meaningful tree. Then, a tree building method should be applied. There are two general ways for phylogenetic tree construction. Distance-based methods apply distance metric to the sequences then use a clustering algorithm to infer their relatedness. Character-based methods treat each substitution separately instead of limiting the variations between sequences to a single distance value [37]. All tree-building methods rely on statistical models that describe the patterns of amino acid replacement. [38].

The unweighted pair group method with arithmetic means (UPGMA) [39] and the neighbor joining (NJ) [40] are commonly used in distance-based methods. UPGMA is a simple hierarchical clustering method. UPGMA finds the most closely related pairs of sequences according to the distance matrix. Those two pairs are then clustered and grouped together as a single internal node with the average distance between them being the branch lengths. Then, the next closely related pair of sequences (or sequence and cluster group) is identified and the same process continues until all sequences are included in the tree. UPGMA assumes that the rate of amino acid substitution is constant for all the branches in the tree. This

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

simplifying assumption makes this method less accurate than others such as NJ.

The NJ method does not assume all lineages evolve at the same rate. The algorithm starts with a star-like tree where all nodes are terminal nodes. Then, it modifies the distance matrix in such a way that the distance between each pair reflects the average divergence from all other nodes. Next, the tree links the pairs with the least distance in the modified matrix. Form this point forward, the joined pairs are represented by an ancestral node and their terminal nodes will not be considered. The distance between this ancestral node and all other terminal nodes is recalculated, and the process continues until the last two nodes that are connected together in the initial start-like tree remain.

The most commonly used character-based methods are the maximum parsimony [41], maximum likelihood (ML) [42] and Bayesian methods. The character-based method uses the aligned sequences to infer the tree instead of limiting the inference to a single distance matrix. This method is also called optimality-based method, where an optimality criterion is used to measure a tree fit to data, and the tree with the best score is the estimated tree [43].

Maximum parsimony infers a phylogenetic tree by minimizing the branch length, so the tree with minimum number of changes is the best tree. The first step is to identify the informative sites. The site is informative if there are at least two different amino acid residues each of which is in at least two sequences in the alignments. Then, tree construction applies to only those informative sites. If a small number of sequences are evaluated, all possible trees are constructed. Otherwise, heuristic methods are applied to reduce the number of constructed trees so that trees that are unlikely to contain the shortest branches are skipped. Finally, the method counts the number of changes over all the informative sites in all trees and chooses the tree with the minimum number. Because this method assumes that the rate of change in all sequences is the same, it suffers from an artifact called long branch attraction [44]. Long branch attraction refers to a situation where sequences under study have different evolution rates, which cause the rapidly evolving sequences to be grouped together even if they are not truly closely related.

The ML method estimates the tree topology, or the branch lengths that maximize the probability of observing the sequences under study. Because constructing all possible trees may not be computationally feasible, heuristic algorithms need to

be applied. All tree construction methods rely on heuristic searches to find the best tree topology. One of the computationally tractable maximum likelihood methods is implemented in the Tree-Puzzle method [45]. Briefly, the Tree-Puzzle method has three main steps. First, all possible quartets that can be formed from the evaluated sequences are constructed. Then, intermediate trees are computed by repeatedly combining sequences to the already computed subtree. Finally, a majority rule consensus is computed using intermediate trees from the previous step. The Bayesian method extends to likelihood methods to use Bayesian statistics. Bayes theorem uses prior probability to compute the posterior distribution of trees with high likelihood given the dataset. In addition, the Monte Carlo Markov Chain (MCMC) is used to estimate the posterior probability distribution [38].

Each of the distance-based and character-based methods has its advantages and weaknesses. A summary of the reviewed methods can be found in Table 2.

As mentioned previously, the quality of phylogenetic tree is directly related to the used MSA algorithm. Once the proper MSA algorithm is used, and the phylogenetic tree is constructed it is usually useful to assess how the predicted relations are supported by the data in the MSA. Bootstrap analysis is the most common way used to assess the robustness of the constructed tree. In the bootstrap method, the data is resampled with replacement by randomly choosing columns of the MSA. The resampled data is the same size as that of the original data set. Then, a new tree is generated from the resampled data set. This process is repeated until usually between 100 and 1000 new trees are constructed [36] [38]. Finally, the bootstrap trees are compared with the original tree and bootstrap values are assigned to the original tree. For example, if an inner node with two children A and B is given a bootstrap value of 90%, this means that in 90% of the bootstrap trees A and B were siblings.

2.3 Protein Composition

Protein sequences have a lot 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.* [46], who found that

Method	Advantages	Cautions
UPGMA	-Fast, simple -Capable of handling large data sets	-Large sequence information is lost; ancestral sequences at internal nodes cannot be inferred -Assumes constant rate of evolution
NJ	-Fast, simple -Handles data with different evolution rates. -Commonly used	- Large sequence information is lost, ancestral sequences at internal nodes cannot be inferred
Maximum Parsimony	-Individual characters are considered in building a tree -Ancestral sequences at internal nodes can be inferred -Faster than other character-based methods	-Branch attraction problem -Assumes constant rate of evolution
ML	-Individual characters are considered in building a tree -Ancestral sequences at internal nodes can be inferred -More accurate than NJ, maximum parsimony -Handles data with different evolution rates.	-Very CPU intensive and extremely slow
Bayesian Method	-Same as ML, more accurate	-More computationally intensive

Table 2: Phylogenetic tree-construction methods

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 [47] [48] [49], subcellular localization [50], whether a transmembrane protein acts as a channel/pore, electrochemical potential-driven transporters, or primary active transporters [15].

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

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 i^{th} 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 i^{th} amino acid.

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 i^{th} 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 i^{th} and j^{th} amino acid.

3. Pseudo-Amino Acid Composition (PseAAC)

PseAAC, which was proposed in 2001 by Chou [51] where he proved that it shows 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:

$$R_1 R_2 R_3 R_4 ... R_L \tag{5}$$

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:

$$\begin{aligned} \Theta(R_i, R_j) = \frac{1}{3} \big\{ [H_1(R_j) - H_1(R_i)]^2 + [H_2(R_j) - H_2(R_i)]^2 \\ + [M(R_j) - M(R_i)]^2 \big\} \end{aligned} \tag{7}$$

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 [52] ; $H_2^{\circ}(R_i)$ 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 [53]. 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:

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

where f_i is the normalized occurrence frequency of the of the ith amino acid in the protein sequence, θ_j is the j^{th} 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 [51] is .05.

3 Projects

3.1 Introduction

Transmembrane proteins classification can be according to many different criteria. For example, one could classify the transmembrane proteins based on their structure, sequence similarity, or substrate specificity. In this project, our aim is to understand some of the proposed solutions, discuss their limitations and conclude on what may be done to improve the overall classification. We chose two papers that used completely different techniques: phylogenetic and amino acid composition. The first paper by Struck [54] was chosen mainly because it is the most recently published paper in the context of transmembrane transporters. The second selected paper by Helms *et al.* [16] shows very promising results using amino acid compositions.

We shall divide this project into three subprojects. The first subproject discusses how can we distinguish transmembrane proteins from other proteins. The second subproject focuses on classifying transmembrane proteins according to their TCDB families using sequence similarity; Struck [54] paper was used as a guideline to achieve this. Finally, the third subsection applies amino acid compositions to find the proteins substrate specificity; Helms *et al.* [16] paper severed as guide to achieve this.

3.1.1 Hardware Specifications

All programs were run on a MacBook pro OS X Yosemite (version 10.10) with Intel Core i7 @ 2.3 GHz processor, 8 GB 1600 MHz DDR3 RAM and 250 GB HD storage.

3.2 Subproject 1: Distinguishing Transmembrane Proteins

The phospholipids of a cell membrane are arranged in a double layer where the polar, hydrophilic (water loving) phosphate heads face the outer part of the membrane and the hydrophobic, nonpolar tails are buried in the membrane interior. Since the transmembrane proteins are the integral proteins 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 transmembrane segments (TMS) can have either α-helical or β-barrel structures; α-helical are the most common structures, and the only known occurrence of β-barrels TMS is in the outer membrane of Gram-negative bacteria [55].

3.2.1 Materials and Methods

We used TMHMM [56] and HMMTOP [57] and compared their performances in finding the number of TMS. A protein is classified as a transmembrane protein if it has at least one predicted TMS. Otherwise, it was classified as a non-membrane protein.

We limited the testing to only one well-studied organism Arabidopsis thaliana. We searched SWISS-PROT [58] using the following search queries. For transmembrane proteins:

```
annotation:(type:transmem) AND reviewed:yes AND organism:
 "Arabidopsis thaliana (Mouse-ear cress) [3702]"
```

We retrieved 3,169 sequences, all of which were used as data. For non transmembrane proteins:

```
NOT annotation:(type:transmem) AND reviewed:yes AND organism:
 "Arabidopsis thaliana (Mouse-ear cress) [3702]"
```

We retrieved 11,102 sequences, and randomly sampled (without replacement) 3200 as our data.

We run HMMTOP and TMHMM on their on-line servers, choosing "one line per protein" option. This option will output for each protein,its length, the number of TMS and the orientation of the first residue relative to the membrane (inside or outside) and the location of predicted helices. Then, we input the resulted text file to R program that classify a protein as transmembrane protein or not according to the number of predicted TMS. finally the program calculates the performance of both HMMTOP and TMHMM. Three statistical measures were considered to measure the performance sensitivity, specificity, and accuracy, which

were calculated in a conventional way:

$$Sensitivity = \frac{TP}{TP + FN} \tag{11}$$

$$Specificity = \frac{TN}{TN + FP} \tag{12}$$

$$Accurecy = \frac{TP + TN}{TP + FN + TN + FP} \tag{13}$$

Where TP is the true positive, FN is the false negative, TN is the true negative and FP is false positive.

3.2.2 Results and Discussion

The first step before classifying transmembrane proteins is to determine whether a protein is a transmembrane protein or not. The existence of transmembrane segments in the protein structure can be used as an indication that the protein is in fact a transmembrane protein. The main limitation here is that the tools that detect the number of the transmembrane segments are not perfect and can make a wrong prediction. For example, TMHMM and HMMTOP are the most accurate methods when it comes to finding the number of transmembrane segments. It has been noticed before that TMHMM is the most selective method for avoiding false positive predictions [59]. To test this, both TMHMM and HMMTOP were used on our testing data. As shown in Table 3, this claim appears to be true.

We have also noticed that the actual number of transmembrane proteins is better detected using HMMTOP. This may suggest that one can use TMHMM as a first step in filtering the transmembrane proteins and then apply HMMTOP to get the number of TMS. It is also important to note that these methods address the transmembrane helices and not the β-barrels structures. While using such methods it makes sense to use a model organism such as Arabidopsis thaliana, since β-barrels are only found in the outer membrane of Gram-negative bacteria. An extension to incorporate the transmembrane β-barrel topology prediction is desirable if we need to generalize the solution.

		Actual	
		T	N
Predicted	T	3070	1036
	N	99	2164
Sensitivity		0.96	
Specificity		0.67	
Accuracy		0.81	
HMMTOP			

		Actual	
		T	N
Predicted	T	2885	253
	N	284	2947
Sensitivity		0.91	
Specificity		0.92	
Accuracy		0.92	
TMHMM			

T: Protein with TMS, N:Protein without TMS

TMHMM is the most selective method for avoiding false positive in TMS predictions. HMMTOP has more false positives than TMHMM and there is great imbalance between sensitivity (true positive rate) and specificity (true negative rate).

Table 3: TMHMM and HMMTOP perdition comparison

3.3 Subproject 2: Finding Protein's Family

The premise behind using phylogenetic trees to infer the function is that sequences with high similarity usually belong to the same family and thus have the same function. Phylogenetic trees provide visualization of the evolutionary history of molecular sequences.

Struck [54] attempts to map putative amino acid transporters with unknown transport activity from rust fungi to the functionality of well-characterized homologous proteins of other fungal species using phylogenetic inference. Rust fungi are specialized parasites that spend all their life in the host plant tissue, and are considered as a significant cause of plant diseases.

3.3.1 Materials and Methods

The first step was to replicate the tree following the same methods used in Struck [54] paper. Struck [53] aligned the protein sequences using ClustalW and built the phylogenetic tree using the Neighbor-Joining method with Poisson correction model and pairwise deletion of the gaps. All of this was done using Molecular Evolutionary Genetics Analysis (MEGA) software version 6.0 [60]. Also, TCDB classification was used to classify the proteins (see Section 1.3). The resulted tree was assessed using 1000 bootstrap replicas.

The dataset was obtained from Uniprot using the Uniprot ID provided in the trees (see Figures 4 and 5). The blue highlighted sequences are 13 previously characterized proteins: as members of the Yeast Amino Acid Transporter (YAT) family (TC# 2.A.3) the ascomycetous S. cerevisiae PUT4 protein has been selected together with the ecto- mycorrhizal permeases GAP1 of Hebeloma cylindrosporum and AAT1 of Amanita muscaria and the U. fabae amino acid permeases AAT1, AAT2,and AAT3. The yeast methionine specific transporters MUP1 and MUP3 were selected as members of the L-type Amino Acid Transporter (LAT) family (TC#2.A.3.8). The -aminobutyric acid transporters UGA4 of S. cerevisiae and gabA of Emericella nidulans were selected as members of the Amino Acid/Choline Transporter (ACT) family (TC# 2.A.3.4). From the the Amino Acid/Auxin Permease (AAAP) superfamily (TC# 2.A.18) the vacuole amino acid transporters of S. cerevisiae AVT5 and AVT7 have been selected and the Arabidopsis thaliana permeases AAP1 and AAP2.

A total 60 transporter proteins of P. graminis-tritici (21), P. triticina (14), M. lini (4), M. larici-populina (16), the fern rust fungus Mixia osmundae (5) were used as putative amino acid transports.

After that, we tried to improve the resulting tree by using different MSA algorithms including Clustal Omega 1.2, MAFFT 7.2, T-COFFEE 11.0 and TM-COFFEE 11.0. In addition, we applied Transitive Consistency Score 11.0 (TCS) [61] to the alignment. 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 (blue) to 9 (red), where 0 is extremely uncertain and 9 is very reliable, as shown in Figure 3.

It has been shown that the highly reliable portions are the most informative when constructing a phylogenetic tree and the most likely to be structurally correct regions [61]. So, by filtering the alignment in such a way that only the highly reliable columns are presented, a more accurate phylogenetic tree is constructed. In our work, we filtered out the columns that have a reliability index value below 3.

Branches with bootstrap values below 60 are considered weakly supported.

Finally, we used figTree1.4.2 [62] to produce publication quality tree figures .

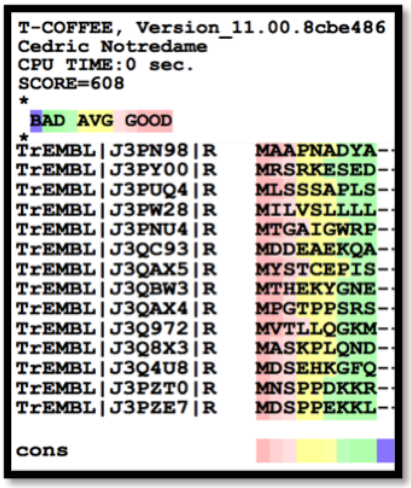

Figure 3: A sample of TCS output

3.3.2 Results and Discussion

The replicated tree from the Struck [53] paper is shown in Figure 4. This tree divides the protein sequences into two superfamilies. The first superfamily is The Amino Acid-Polyamine-Organocation (APC) superfamily (TC# 2.A.3), which is further divided into three subfamilies. The pink represents the YAT, the yellow highlights the LAT family, and the orange highlights the ACT family. The second superfamily, highlighted in turquoise, is the AAAP superfamily. This division is based on the formerly characterized proteins. So any protein with unknown function that belong to a certain family is expected to follow the same behavior of the family based on sequence similarity.

The tree in Figure 4 appears to have many low bootstrap values (pink-circled), which suggests that in around half of the bootstrap replicas the corresponding branches were not supported. Hence, they are not reliable.

This is especially concerning in the clade that contains LAT (yellow) and ACT (orange) families, because how can we be confident in the classification when almost half of the bootstrap replicas did not support this clade?

Our aim is to build a more robust and consistent tree than what was suggested in the paper. After experimenting with many MSA algorithms, we found that TM-COFFEE gives the highest TCS score when compared to other alignments. For example, the difference in TM-COFFEE and ClustalW TCS scores are displayed

in Table 4. The difference was assessed by Student t-test (two tailed,independent) to be statistically significant (P value = 0.0137).

Using TM-COFFEE while filtering the unreliable columns and trusting only branches with bootstrap values of at least 60% will produce a tree illustrated in Figure 5. The families (YAT,LAT,ACT) and the superfamily AAAP stayed the same as ClustalW tree (Figure 4). The difference was that in ClustalW tree LAT and ACT families had common ancestor with each other, and that common ancestor was shared with YAT family, while here YAT,LAT, and ACT families share only one common ancestor. This difference is not a significant in this example, but this is not always the case.

The most noticeable observation from this project is that even transporters that transport the same substrate (amino acid in this case) may have a very low sequence identity (as low as 11%) and belong to diverse superfamilies, and classifying the substrates merely based on their sequence similarity may not be optimal. So the question here is what do those transporters have in common? To answer this question, we applied many bioinformatics techniques such as MEME [63] for motif discovery and JDet [64] for finding specificity-determining positions (SDP) without a useful output. This suggests that functionality signals based on the sequence of amino acids and their relative order is not clear and further research should put into finding the mutual features of the same substrate transporters.

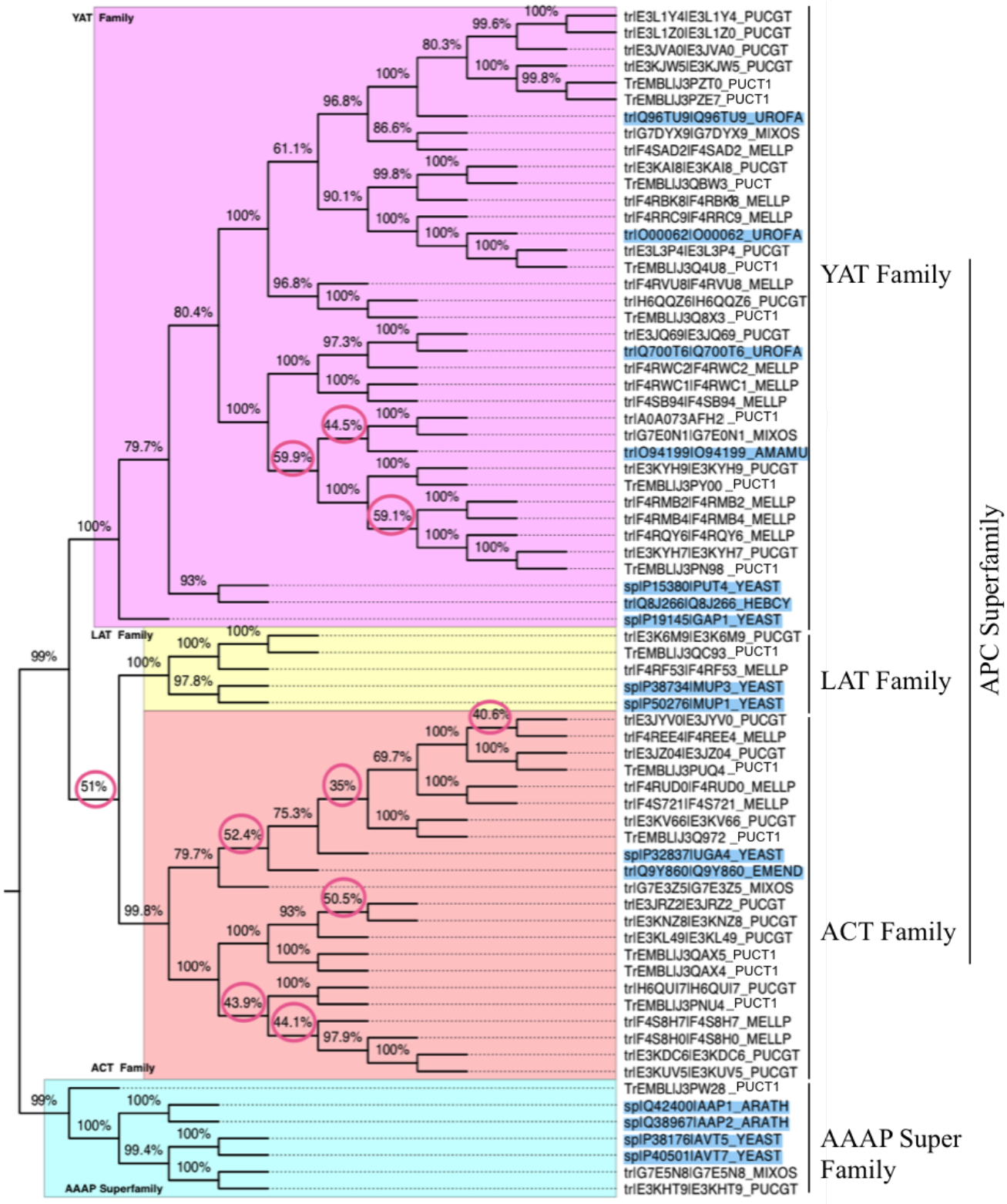

Figure 4: Neighbor Joining tree of putative amino acid transporters using ClustalW

A replicated Tree from Struck [54] paper that has many weakly supported (< 60%) branches (pink circled). The blue highlighted transporters are formerly characterized. Bootstrap values are calculated from 1000 replicas.

Sequences are designated with Uniprot ID and five letter species names: ARATH, Arabidopsis thaliana; EMEND, Emericella nidulans; HEBCY, Hebeloma cylindrosporum; MELLI, Melampsora lini; MELLP, M. larici-populina; MIXOS, Mixia osmundae; PUCGT, Puccinia graminis-tritici; PUCT1, P. triticina; UROFA, Uromyces fabae; YEAST, Saccharomyces cerevisiae.

UniprotID	TM-COFFEE	ClustalW
E3L1Y4	70	67
E3L1Z0	72	67
E3JVA0	67	60
E3KJW5	69	65
Q96TU9	70	66
G7DYX9	69	65
F4SAD2	68	65
E3KAI8	70	67
F4RBK8	70	67
F4RRC9	69	65
O00062	69	65
E3L3P4	70	65
F4RVU8	70	66
H6QQZ6	67	62
E3JQ69	70	66
Q700T6	70	66
F4RWC2	69	65
F4RWC1	70	66
F4SB94	70	66
A0A073	69	66
G7E0N1	69	65
O94199	69	65
E3KYH9	69	65
F4RMB2	71	69
F4RMB4	72	69
F4RQY6	69	66
E3KYH7	69	65
P15380	68	62
Q8J266	68	64
P19145	69	64
E3K6M9	56	46
F4RF53	54	45
P38734	48	40
P50276	49	41
E3JYV0	58	52
F4REE4	57	51

UniprotID	TM-COFFEE	ClustalW
E3JZ04	57	51
F4RUD0	53	48
F4S721	54	49
E3KV66	50	45
P32837	50	47
G7E3Z5	55	47
Q9Y860	56	53
E3JRZ2	54	49
E3KNZ8	56	52
E3KL49	56	51
H6QUI7	52	28
F4S8H7	55	50
F4S8H0	56	51
E3KPAAC6	54	51
E3KUV5	54	50
P38176	31	2
P40501	32	4
G7E5N8	28	4
E3KHT9	21	8
Q42400	24	5
Q38967	22	3
J3PN98	68	64
J3PY00	69	66
J3PUQ4	55	50
J3PW28	25	3
J3PNU4	58	52
J3QC93	56	47
J3QAX5	56	51
J3QBW3	66	62
J3QAX4	56	50
J3Q972	57	52
J3Q8X3	70	66
J3Q4U8	64	59
J3PZT0	67	63
J3PZE7	68	64

TM-COFFEE TCS score is always larger than ClustalW TCS score. The difference in TCS scores was assessed by Student t-test (two tailed,independent) to be statistically significant (P value = 0.0137). This is an example that shows that TM-COFFEE has higher reliability scores when compared to other aligners, which reflects its higher accuracy in aligning transmembrane proteins.

Table 4: A comparison between TM-COFFEE and ClustalW TCS Scores.

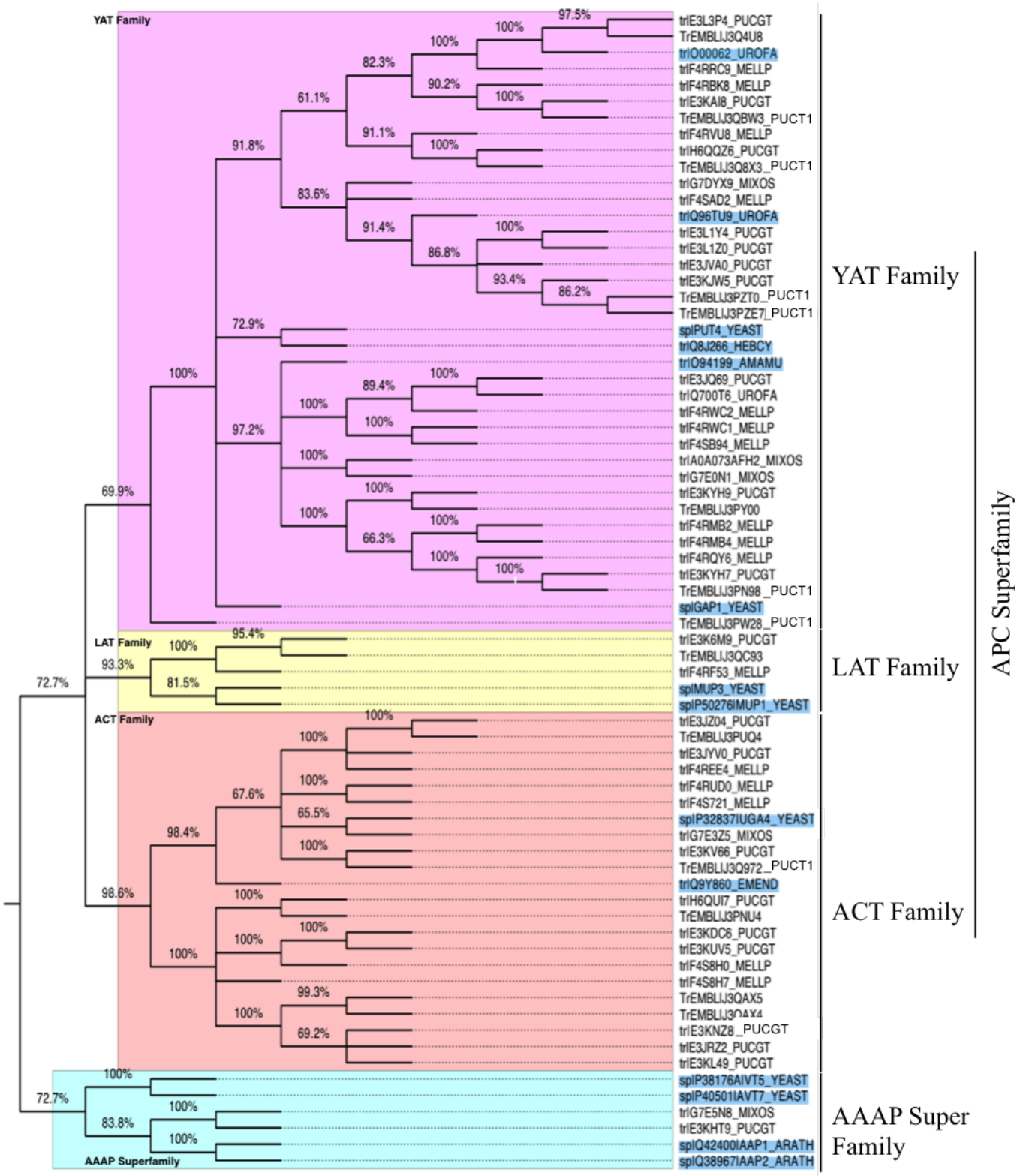

Figure 5: Neighbor Joining tree of putative amino acid transporters using TM-COFFEE

This figure shows no weakly supported branches using TM-COFFEE. The blue highlighted transporters are formerly characterized. Bootstrap values are calculated from 1000 replicas. TM-Coffee is used with uniref50-TM dataset for homology extension.

Sequences are designated with Uniprot ID and five letter species names: ARATH, Arabidopsis thaliana; EMEND, Emericella nidulans; HEBCY, Hebeloma cylindrosporum; MELLI, Melampsora lini; MELLP, M. larici-populina; MIXOS, Mixia osmundae; PUCGT, Puccinia graminis-tritici; PUCT1, P. triticina; UROFA, Uromyces fabae; YEAST, Saccharomyces cerevisiae.

3.4 Subproject 3: Finding Substrate Specificity

Helms *et al.* [16] used a combination of different amino acid composition methods (see Section 2.3) in addition to amino acid conservation with homologues sequences to detect different substrate specificity. The investigations were done on Arabidopsis thaliana transmembrane proteins because of the availability of well-annotated substrates for this plant. In this project, Helms *et al.* [16] paper was used as a guideline to classify the substrate specificity of a given protein sequence; some details were changed to fit our computational power or to improve the overall classification.

3.4.1 Materials and Methods

Helms *et al.* [16] considered four different substrate classes: amino acids, oligopeptides, phosphates, and hexoses. The interesting remark here is that not all the substrates belong to the same TCDB family and they do not necessarily have high sequence similarity. The general information about the data can be found in Table 5. Details are found in the Appendix.

Substrate class	**Set size**	**Number of TCDB families**	**Families TCDB ID**
Amino acid	15	2	2.A.18, 2.A.3
Oligopeptide	17	2	2.A.67, 2.A.17
Phosphate	15	5	2.A.1.9, 2.A.1.14 2.A.20, 2.A.29, 2.A.7
Hexose	15	2	2.A.1.1, 2.A.123

Table 5: Substrate specificity Dataset

The implementation was done using R programming language. There are two major steps needed for the substrate specificity classification: data preparation and classification.

- **Data Preparation**

  AAC, PAAC and PseAAC ($\lambda = 30$) are implemented as described in Section 2.3 and used to find the compositions of all sequences in the dataset. In addition, Helms *et al.* [16] incorporated evolutionary information which aided

to a higher overall accuracy than solely using sequence-inclusive information. A method called MSA-AAC is used to achieve this. The first step is to create a local database. We created a local database by combining all sequences from SWISS-PROT that have the transmembrane keyword. Next, for each sequence i in each subset a BLAST (version 2.2.18) search was performed retrieving a maximum of 120 homologous sequences. Then, the retrieved homologous sequences were aligned using MAFFT 7.2.

Afterwards, the aligned sequences were filtered in such a way that sequences with identity below 25% were removed. We added one additional filter TCS 11.0 [61] in the same way we did in Section 3.3.1. Finally, AAC calculations were performed to each sequence in the alignment. Then the mean was considered as the MSA-AAC for that sequence i (See Algorithm 1).

Algorithm 1 MSA-AAC algorithm

```
1: procedure MSA–AAC
2:     locaDB ← Search(SWISS-PROT, keyword="transmembrane")
3:     vector< vector< int >> MSA-AAC=
           new vector < vector< int >> (#seq in Dataset)
4:     for each sequence i ∈ Dataset do
5:         retrievedSeq← BLAST( DB= localDB, query= sequence i,
             maxseq= 120 )
6:         MSA← Align(retrievedSeq,MAFFT)
7:         filteredMSA← TCS(MSA, tcs_column_filter3)
8:         for each sequence z ∈ filteredMSA do
9:             AACvector ← AAC(sequence z)
10:        end for
11:        MSA-AAC[i]← mean(AACvector)
12:    end for
13: end procedure
```

We also included MSA-PAAC, MSA-PseAAC information in the classification, which is done in a similar manner as MSA-AAC, except in the final step, PAAC and PseAAC calculations are performed, respectively.We end up with six feature vectors that represent each type of amino acid composition. A summary of the steps required to prepare the data before classification are

shown in Figure 6.

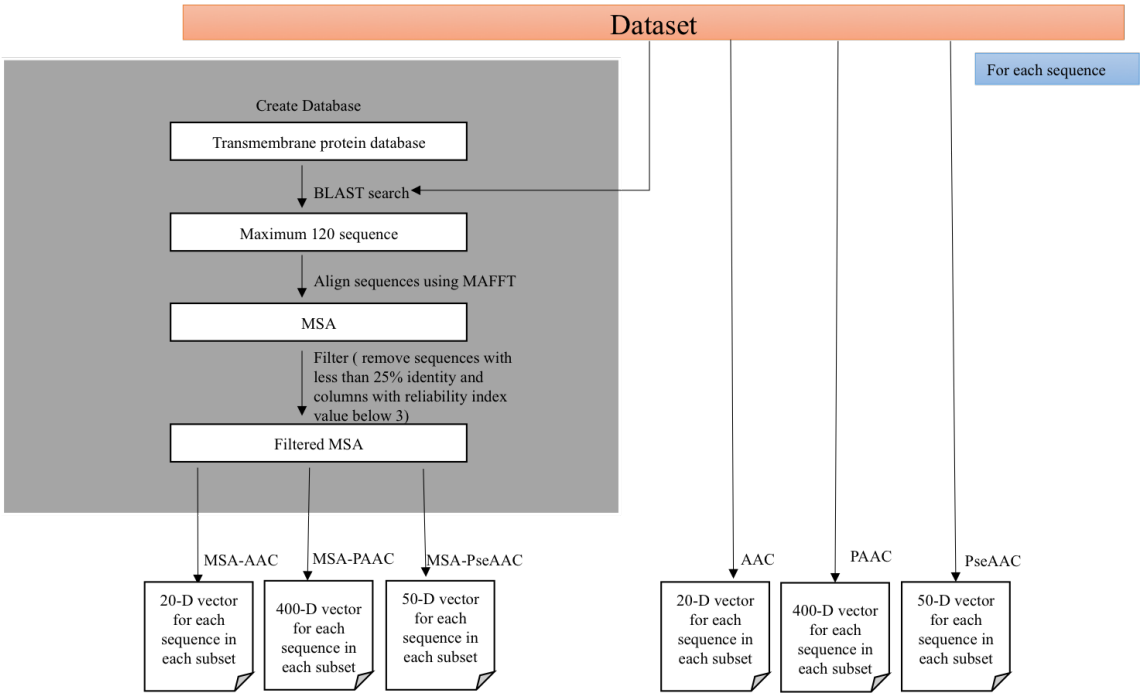

Figure 6: The preparation step for the substrate specificity classification

- **Classification**

  Six K-Nearest Neighbors classifiers ($K = 1$) were built corresponding to the six different types of compositions, AAC, PAAC, PseAAC, MSA-AAC, MSA-PAAC, and MSA-PseAAC, that resulted from the preparation step. Each classifier was used independently from the other classifiers to classify a transmembrane protein sequence to one of the four considered substrates: amino acids, oligopeptides, phosphates, and hexoses.

  The final classification was resolved through a voting system. The voting system works as follows: the substrate that gets the most number of votes is predicted. If the major votes are split between two substrates, one of them is picked randomly. If the votes are split between more than two substrates the prediction is unknown for that protein (see Algorithm 2). Due to the limited available data, the classification performance was estimated using leave-one-out cross-validation (LOOCV). Three statistical measures were considered, sensitivity, specificity, and accuracy (see Eq. 11,12,13)

Algorithm 2 Classification algorithm

```
1: procedure CLASSIFICATION
2:     for each i ∈ {AAC, PAAC, PseAAC, MSA-AAC, MSA-PAAC,   ▷ 1st loop
           MSA-PseAAC} feature vectors do
3:         for each s sequence ∈ feature vector i do          ▷ 2nd loop
4:             remove s from feature vector i
5:             build 1 nearest neighbor classifier using the remaining sequences
6:             predict the substrate class of s using the classifier
7:         end for
   ▷ The 2nd loop computes a vector of predicted classes of the sequences in
   feature vector i
8:     end for
   ▷ The 1st loop computes 6 vectors of predicted classes of the sequences in the
   6 feature vectors
9:     for each s sequence in Dataset do                     ▷ begin voting
10:        finalPrediction[s] ← perform the majority vote of the predicted
               classes of s in the 6 resulting vectors
11:    end for
   ▷ The above loop computes the final vector of predicted classes of all the
   sequences
12: end procedure
```

3.4.2 Results and Discussion

Here we look at the property and the sequence-hidden information rather than directly to the sequence of amino acids to classify transmembrane proteins according to their substrate specificity by using AAC,PAAC and PseAAP. In addition, we incorporated evolutionary information through using MSA in MSA-AAC,MSA-PAAC,and MSA-PseAAC. An example of the different amino acid compositions among different substrates in the data is illustrated in Figure 7 and Figure 8. As we can see, Glycine (G) shows the highest variance and hence it is highly distinctive between the different substrates.

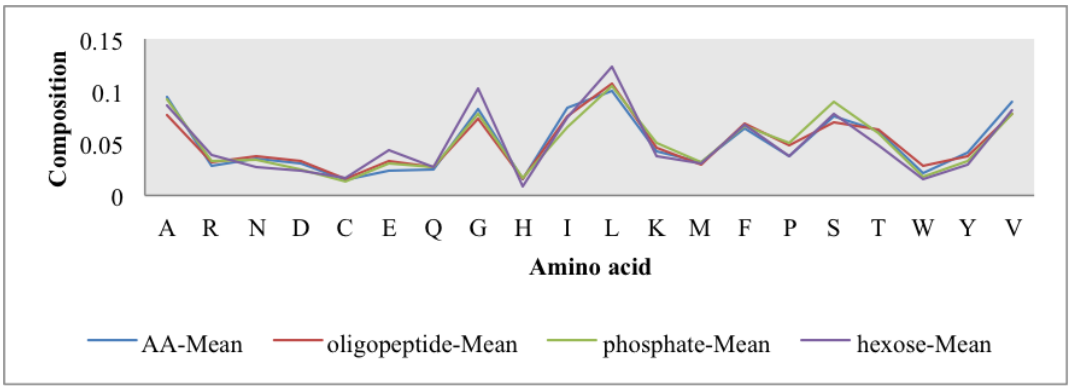

Figure 7: Amino Acid composition of different substrates transporters

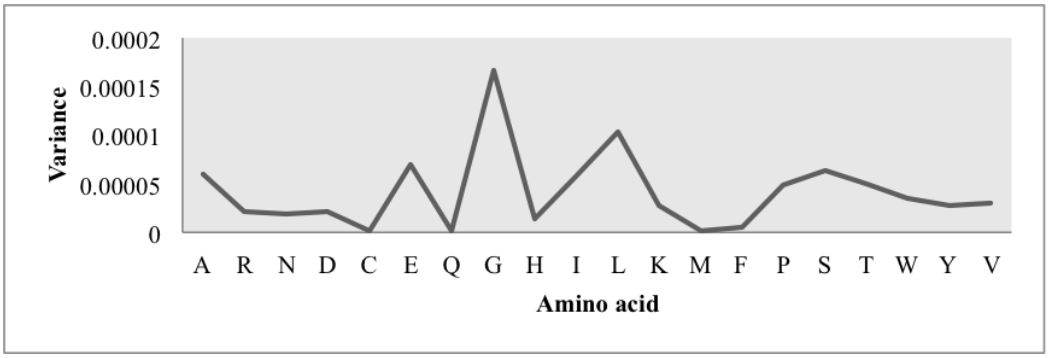

Figure 8: Variance in amino acid composition of different substrate transporters: amino acid, oligopeptide, phosphate and hexose transporters

As for the implementation, we changed a few details than what the paper suggested. First of all, while Helms *et al.* [16] used a non-redundant database [65], which when installed locally consumes more than 18 GB. 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. Then, instead of the 1,000 sequences used in the paper we limited the number of the retrieved sequences to 120 to fit our computational power. In addition, the retrieved homologous sequences were aligned using MAFFT instead of ClustalW because of its proven higher accuracy [17] [34] [35]. The reason that we chose MAFFT rather that TM-COFFEE is because of its more efficient execution, where a sequence can be aligned within 3 minutes in comparison to at least 5 hours in TM-COFFEE. Furthermore, We added the TCS score filtering to th MSA. Moreover, we included MSA-PAAC and MSA-PseAAC with hope to improve the classification, while Helms considered only MSA-AAC. Finally, in the classification we used K-Nearest Neighbors with a voting system, while Helms used a ranking system that is based on Euclidean distance between the amino acid composition of the considered transporter and the mean composition of the each substrate category and combined the results using a cross entropy Monte Carlo (CEMC) method.

We were able to obtain a better accuracy than what was proposed in Helms *et al.* [16] paper in AAC-MSA of all substrate classes except in phosphate we kept the same accurecy (96%) (see Table 6). We could not compare the other MSA algorithms, MSA-PAAC and MSA PseAAC, because they were not included in the paper . We believe that the improved accuracy in MSA was obtained for two reasons, first, including the use of MAFFT rather than ClustalW for the MSA step in the data preparation. Second, the use of TCS to filter the sequences columns that have a reliability index below 3 rather than simply removing the sequences with identity below 25%.

As shown in Table 6, using amino acid compositions alone does not yield high classification accuracy. The high accuracy came from the incorporating evolutionary information using MSA, which suggest that the evolutionary information is key to classifying transmembrane proteins.

Although using the methods suggested in the paper gives an overall accuracy of more than 90% (see Table 7), there is a major concern regarding how well it will perform at a large scale. After all, only 61 transporter proteins were used for training on a single model organism for only four substrate classes. So the question here is can this method be generalized to include more substrate classes for multiple organisms? Can we extend this classification to not only find the general substrate

Subset	Method	Sensitivity		Specificity		Accuracy	
		Helms	X*	Helms	X	Helms	X
Amino Acid	AAC	0.875	0.800	0.875	0.979	0.875	0.889
	PseAAC	0.875	0.733	0.875	0.979	0.875	0.856
	PAAC	0.938	0.733	0.867	0.979	0.903	0.856
	PsePAAC	0.938	-	0.875	-	0.906	-
	MSA-AAC	0.875	1.00	1.00	0.979	0.938	0.989
	MSA-PAAC	-	1.00	-	0.979	-	0.989
	MSA-PseAAC	-	0.933	-	1.00	-	0.966
Oligopeptide	AAC	0.941	0.882	1.00	0.913	0.970	0.897
	PseAAC	0.882	0.882	1.00	0.956	0.939	0.919
	PAAC	0.933	1.00	1.00	0.782	0.968	0.891
	PsePAAC	1.00	-	1.00	-	1.00	-
	MSA-AAC	0.882	1.00	0.667	1.00	0.733	1.00
	MSA-PAAC	-	1.00	-	1.00	-	1.00
	MSA-PseAAC	-	1.00	-	1.00	-	1.00
Phosphate	AAC	0.800	0.750	0.667	0.893	0.733	0.821
	PseAAC	0.800	0.750	0.667	0.957	0.733	0.853
	PAAC	0.933	0.625	1.00	0.957	0.968	0.791
	PsePAAC	0.933	-	1.00	-	0.968	-
	MSA-AAC	0.933	0.937	1.00	1.00	0.968	0.968
	MSA-PAAC	-	0.937	-	1.00	-	0.968
	MSA-PseAAC	-	0.937	-	0.978	-	0.958
Hexose	AAC	0.769	0.733	0.909	0.937	0.833	0.835
	PseAAC	0.769	0.866	0.909	0.854	0.833	0.860
	PAAC	0.769	0.800	1.00	1.00	0.875	0.900
	PsePAAC	0.769	-	1.00	-	0.875	-
	MSA-AAC	0.769	1.00	1.00	1.00	0.875	1.00
	MSA-PAAC	-	1.00	-	1.00	-	1.00
	MSA-PseAAC	-	1.00	-	0.979	-	0.989

* X= Alballa work

Table 6: Detailed substrate specificity performance

Subset	Sensitivity	Specificity	Accuracy
Amino Acid	0.937	1.00	0.968
Oligopeptide	1.00	1.00	1.00
Phosphate	1.00	1.00	0.979
Hexose	0.9375	0.959	0.968
Voting accuracy			0.968

Table 7: Overall substrate specificity voting performance when one model organism (Arabidopsis thaliana) is used

(e.g. amino acid) but also the exact substrate (e.g. Lysine)? Finding adequate answer to those questions is not an easy task. All the made efforts are considered attempts and the research area is far behind finding a generalized solution.

3.5 Future Work

The project here was mainly implemented to enrich our understanding of the current sate-of-art methods. In the future we are eager to find a generalized solution by:

- Extending the substrate groups to include all possible transported substrates
- Finding the exact transported substrate (e.g. Lysine) rather than the general substrate (e.g. amino acid)
- Taking into an account different organisms

4 Conclusion

Transmembrane proteins play extremely important roles in all living cells; yet they are among the least characterized protein owing to their instable features. There is an insistent 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.

This report covers basic biological concepts related to transmembrane proteins and their features. In addition, the important bioinformatics methods that are needed to find the computational solutions are also covered. Furthermore, we have implemented, with modification, two methods to classify transmembrane proteins using two papers as a guide. We have seen that using phylogenetic trees to classify transmembrane proteins according to their TCDB family is not always useful. Since phylogenetic tree relies on sequence similarity to infer the function of the questioned transmembrane proteins, it fails when the homologues proteins have low sequence similarity; which is occasionally the case in transmembrane proteins. We have also found that using protein compositions yield to a better substrate classification. In addition, integrating homology information to the amino acid composition is a key to improving the overall classification performance.

There is still a lot to be done in the area of transmembrane proteins classification, all of the implemented methods and the published papers are in the initial stages and there is not yet a general solution.

References

[1] Wikipedia, "Cell membrane," 2015, [Online; accessed 10-Nov-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Cell_membrane

[2] 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.

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

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

[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] P. Baldi and S. Brunak, "Proteins and proteomics," in *Bioinformatics: the machine learning approach*, 2nd ed. MIT press.

[9] 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.

[10] 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 (CIBCB), 2015 IEEE Conference on*. IEEE, 2015, pp. 1–9.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] R. C. Edgar and S. Batzoglou, "Multiple sequence alignment," *Current opinion in structural biology*, vol. 16, no. 3, pp. 368–373, 2006.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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, no. 1, p. 539, 2011.

[23] 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," *Algorithm Mol Biol*, vol. 5, p. 21, 2010.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

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

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

[40] 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.

[41] R. Eck and M. O. Dayhoff, *Atlas of protein sequence and structure*. Minneapolis: Silver Spring.

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

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

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] C. Struck, "Amino acid uptake in rust fungi," *Frontiers in plant science*, vol. 6, 2015.

[55] 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.

[56] 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.

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

[58] 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.

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

[60] 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.

[61] 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*, p. msu117, 2014.

[62] A. Rambaut, "figTree," http://tree.bio.ed.ac.uk/software/figtree/, 2014.

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

[64] 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.

[65] NCBI, "NCBI ftp site," [Online; accessed 19-Nov-2015]. [Online]. Available: ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nr.gz

[66] Uniprot, "Uniprot," [Online; accessed 19-Nov-2015]. [Online]. Available: ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nr.gz

Appendix

Substrate Specificity Sequence Details

The detailed information about the used sequences in Subproject 3 based on Helms *et al.* [16] paper are found in this section.

Substrate	Uniprot ID	TCDB Family
Amino acid	P92934	2.A.18 The amino acid/auxin permease (AAAP)
	P92961	2.A.18 The amino acid/auxin permease (AAAP)
	P92962	2.A.18 The amino acid/auxin permease (AAAP)
	P92962	2.A.18 The amino acid/auxin permease (AAAP)
	Q38967	2.A.18 The amino acid/auxin permease (AAAP)
	Q39134	2.A.18 The amino acid/auxin permease (AAAP)
	Q42400	2.A.18 The amino acid/auxin permease (AAAP)
	Q8GUM3	2.A.18 The amino acid/auxin permease (AAAP)
	Q9FN04	2.A.18 The amino acid/auxin permease (AAAP)
	Q9SF09	2.A.18 The amino acid/auxin permease (AAAP)
	Q9SJP9	2.A.18 The amino acid/auxin permease (AAAP)
	Q9ZU50	2.A.3 The amino acid-polyamine-organocation (APC)
	Q84MA5	2.A.3 The amino acid-polyamine-organocation (APC)
	Q9FFL1	2.A.3 The amino acid-polyamine-organocation (APC)
	Q8W4K3	2.A.3 The amino acid-polyamine-organocation (APC)
Oligpeptide	O04514	2.A.67 The oligopeptide transporter (OPT)
	O23482	2.A.67 The oligopeptide transporter (OPT)
	O82485	2.A.67 The oligopeptide transporter (OPT)
	Q9FG72	2.A.67 The oligopeptide transporter (OPT)
	Q9FJD1	2.A.67 The oligopeptide transporter (OPT)
	Q9FME8	2.A.67 The oligopeptide transporter (OPT)
	Q9SUA4	2.A.67 The oligopeptide transporter (OPT)
	Q9T095	2.A.67 The oligopeptide transporter (OPT)
	Q9FJD2	2.A.67 The oligopeptide transporter (OPT)
	P46032	2.A.67 The oligopeptide transporter (OPT)
	Q05085	2.A.17 The oligopeptide transporter (OPT)
	Q9LFX9	2.A.17 The proton-dependent oligopeptide transporter (POT/PTR)
	Q9LSE8	2.A.17 The proton-dependent oligopeptide transporter (POT/PTR)
	Q9M172	2.A.17 The proton-dependent oligopeptide transporter (POT/PTR)
	Q9M174	2.A.17 The proton-dependent oligopeptide transporter (POT/PTR)
	Q9M175	2.A.17 The proton-dependent oligopeptide transporter (POT/PTR)
	Q9SZY4	2.A.17 The proton-dependent oligopeptide transporter (POT/PTR)

Substrate	Uniprot ID	TCDB Family
Phosphate	O48639	2.A.1.9 The phosphate: H+ symporter (PHS)
	Q8VYM2	2.A.1.9 The phosphate: H+ symporter (PHS)
	Q96243	2.A.1.9 The phosphate: H+ symporter (PHS)
	Q9S735	2.A.1.9 The phosphate: H+ symporter (PHS)
	Q96303	2.A.1.14 The anion:cation symporter (ACS)
	Q9FKV1	2.A.1.14 The anion:cation symporter (ACS)
	O82390	2.A.1.14 The anion:cation symporter (ACS)
	Q3E9A0	2.A.1.14 The anion:cation symporter (ACS)
	Q38954	2.A.20 The inorganic phosphate transporter (PiT)
	Q7DNC3	2.A.29 The mitochondrial carrier (MC)
	Q9FMU6	2.A.29 The mitochondrial carrier (MC)
	Q9M2Z8	2.A.29 The mitochondrial carrier (MC)
	Q8H0T6	2.A.7 The drug/metabolite transporter (DMT)
	Q8RXN3	2.A.7 The drug/metabolite transporter (DMT)
	Q94B38	2.A.7 The drug/metabolite transporter (DMT)
Hexose	O04036	2.A.1.1 The sugar porter (SP)
	P23586	2.A.1.1 The sugar porter (SP)
	Q0WWW9	2.A.1.1 The sugar porter (SP)
	Q39228	2.A.1.1 The sugar porter (SP)
	Q56ZZ7	2.A.1.1 The sugar porter (SP)
	Q8L6Z8	2.A.1.1 The sugar porter (SP)
	Q9C757	2.A.1.1 The sugar porter (SP)
	Q9FMX3	2.A.1.1 The sugar porter (SP)
	Q2V4B9	2.A.1.1 The sugar porter (SP)
	Q6AWX0	2.A.1.1 The sugar porter (SP)
	Q8GW61	2.A.1.1 The sugar porter (SP)
	Q8GXR2	2.A.1.1 The sugar porter (SP)
	Q9LNV3	2.A.123 The sweet; PQ-loop; saliva; MtN3 (Sweet)
	Q8L9J7	2.A.123 The sweet; PQ-loop; saliva; MtN3 (Sweet)
	Q9SMM5	2.A.123 The sweet; PQ-loop; saliva; MtN3 (Sweet)