ON PREDICTING TRANSMEMBRANE PROTEINS FUNCTION

By

Munira Alballa

SUBMITTED AS PHD PROPOSAL REPORT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY
AT
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC
SEPTEMBER 2018

Contents

List of Figures

List of Tables

Chapter I

Data collection

One of the main challenges encountered in building a membrane proteome-wide system is the lack of gold-standard databases. For our problem, we need to define two distinct datasets. The first dataset includes data on membrane proteins with their different types. The second includes data on transporters with their substrate specificities. In this chapter, Section 1 highlights the challenges and illustrates the inconsistencies between the different biological databases. Section 2 and 3 define the main bioinformatics ontologies and databases we used to build our datasets. Next, Section 4 describes the process of building the first database, which contains data about different membrane protein types. Section 5 then outlines the process of building the second database of transporters substrate categories and describes our ontology-based, automated substrate annotation. Finally, Section 6 concludes the chapter.

1 Challenges

Challenges are encountered with respect to multiple aspects of the data-collection process. All of these problems stem from the lack of standardized annotation across different databases. We can encapsulate the issues in three main points: The first is the inconsistency of gold-standard databases. For example, we examined a transporter classification database (TCDB) that contains experimentally characterized proteins (see Section 3.2) and a Swiss-Prot that contains the worldwide primary database of well-annotated and manually inspected data (see Section 3.1). We found that, of 17,000 entries in TCDB, 6,710 are also in Swiss-Prot and only 3,292 of these are annotated with transporter related GO molecular function (GO:0005215 transporter activity). This indicates that Swiss-Prot annotations are stricter than those of TCDB.

The second point is that, at the level of transporter substrate prediction, the ultimate goal is to predict the exact substrate specificity the transporter transports across the membranes. However, this is not obtainable given the current state-of-art because of the lack of well-annotated transporter data. This fact led us to predict the general category (e.g., amino acid) rather than the exact substrate (e.g., arginine). Still, no universally defined sets of gold-standard substrate categories are used in prediction. Researchers are using their own subsets of substrate classes. As the literature review indicates (see Section **??**), some authors [1] 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 [2] and [3]) include oligopeptides —i.e., a few amino acids linked in a polypeptide chain. Others(Chen *et al.* [1] and Mishra *et al.* [4]) elect to incorporate protein/mRNA, which consists of one or more polypeptides with at least 50 amino acids. Still others (Barghash *et al.* [5]) completely discount the protein or oligopeptide category. This raises a challenge when building a database of transporter substrate classes: What substrates should we include in our dataset?

The third point is that all of the substrate specificity datasets (see Section **??**) rely on the manual curation of the biological function annotation (assigning substrate category to a protein sequence). Aside from the fact that manual curation can be extremely time consuming, it also is subject to expert opinion. The details and decisions are often not provided, which makes it difficult to replicate or rebuild the same dataset. Thus, we need to find a way to make this assignment automatic, standardized, easily scalable, and flexible so as to include more or fewer substrate classes.

2 Ontologies and classifications

2.1 Enzyme Classification

The Enzyme Commission (EC) number is a standardized numerical classification scheme for enzymes which describes the chemical reaction catalyzed by an enzyme. The EC numbers are assigned by the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology. Each enzyme code consists of the letters EC followed by four digits separated by periods: e.g., EC X.Y.Z.W. The first digit denotes the general type of reaction catalyzed by the enzyme; it ranges from one to six (see Table 1). The next three digits further define and specify the reaction type.

Class	**Name**	**Reaction catalyzed**
EC 1	Oxidoreductases	Oxidation\reduction reactions
EC 2	Transferases	Transfer of a chemical group from one molecule to another
EC 3	Hydrolases	Formation of two products from a substrate by hydrolysis
EC 4	Lyases	Non-hydrolytic addition or removal of groups from substrates
EC 6	Isomerases	Intramolecule rearrangement
EC 6	Ligasess	Joining of two molecules

Table 1: Top-level reaction classes in enzyme commission numbers

2.2 Chemical Entities of Biological Interest

Chemical Entities of Biological Interest (ChEBI) [6] is a database and ontology which contains information about chemical entities. Each entry in the database is classified within the ontology. ChEBI acts as source of unique reliable identifiers for chemicals in annotation. It is commonly used in many bioinformatics databases and as a chemistry component of several ontologies, including the gene ontology [7].

The ChEBI ontology contains three sub-ontologies:

- . A chemical ontology in which entities are classified based on their structural features and properties (e.g., monosaccharid, carboxylic acid, anion).
- . A role ontology in which the entities are classified on the basis of their activities in

chemical or biological systems (e.g., vitamin, drug, enzyme).

- . A subatomic particle ontology in which particles smaller than atoms are classified (e.g., electron, photon, nucleon).

For each entity in the ChEBI database, information is displayed over several tabs. The main tab is the default tab. It displays general information about the entity (ChEBI name, ID, structure, definition) and shows some of the information contained within the ontology. The ChEBI ontology tab shows the ontology information in further details and provides two options for visualizing the ontology: a graph view and a tree view. The graph view represents a visualization of two relationships—*"is a"* (blue links) and *"has part of"* (black links)—between the ontologies of a certain entity (see Fig 1). The tree view shows all of the immediate relations of a given entity in addition to showing all parents within the hierarchy and the immediate children.

2.3 Gene Ontology

The Gene Ontology (GO) [8] project is the largest resource available that provides an ontology of defined terms with which to represent specific gene products. The GO describes functions with respect to three domains: molecular function (MF), biological process (BP) and cellular component (CC). The molecular function concerns the activities of a gene product at the molecular level. Biological processes include the larger processes accomplished by multiple molecular activities. Cellular components include the locations relative to cellular structures in which a gene product performs its function. The ontology is structured as a directed acyclic graph in which each term has specific relationships to one or more other terms in the same domain. The GO terms that refer to chemical entities have fully defined semantic relationships to corresponding chemical terms in ChEBI. This is done to facilitate an accurate and consistent, systems-wide chemical view of the biological representation [7].

For transporters related terms, we have GO term **GO:0006810 transport** in PB which is defined as the processes involved the directed movement of substances (such as macromolecules, small molecules, ions) or cellular components (such as complexes and organelles) into, out of or within a cell, or between cells, or within a multicellular organism by means of some agent

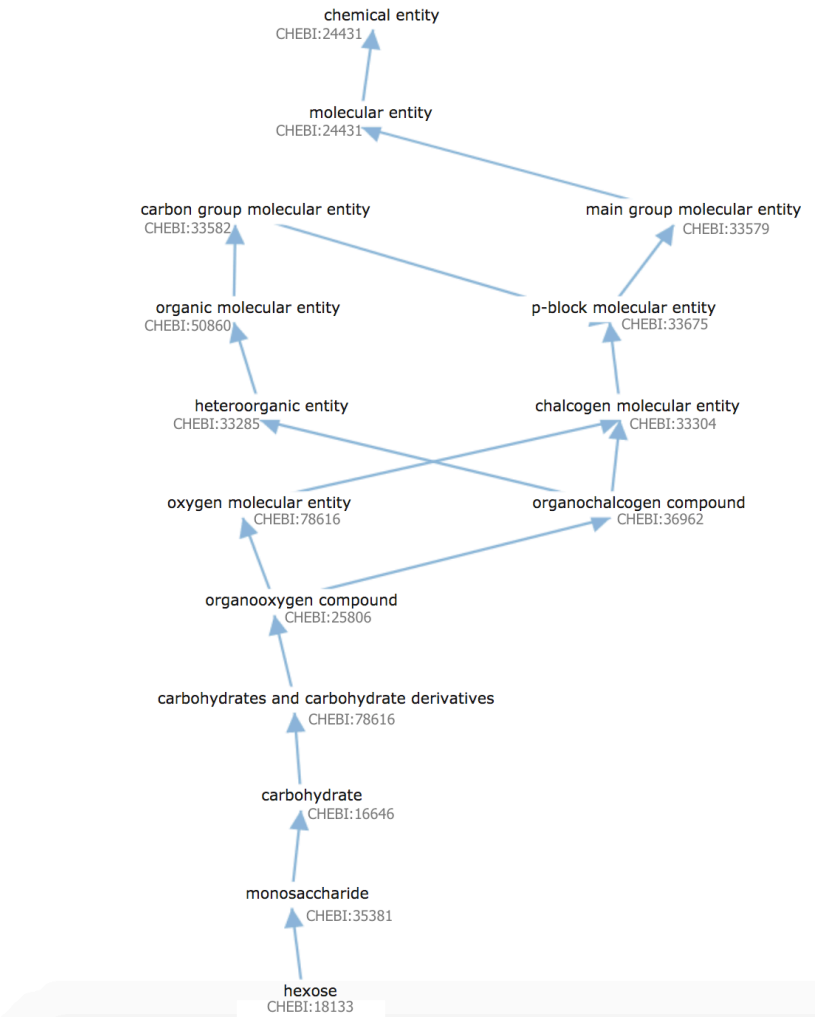

Figure 1: ChEBI Ontology graph view

This figure shows a full graph view of hexose (CHEBI:18133) on the ChEBI ontology, are added to. All of the presented relationships in this case are *"is a"* between the ontologies. By clicking on ontology terms within the view, one can see a definition for that specific term.

such as a transporter, pore or motor protein.

The other GO term is **GO:0005215 transporter activity** in MF that is defined as the function that enables the directed movement of substances (such as macromolecules, small molecules, ions) into, out of or within a cell, or between cells.

2.4 Evidence and Conclusion Ontology

The Evidence Ontology (ECO) is a structured, controlled vocabulary for capturing evidence in biological research. This ontology is used to document evidence-based conclusions derived from investigations. [9].

The current version of ECO contains more than 600 terms arranged in hierarchy with two

high-level classes *"evidence"* (ECO:000000) and *"assertion method"* (ECO:0000217). The *"evidence"* is defined as 'type of information that is used to support an assertion'. The majority of evidence is either experimental (e.g., expression pattern evidence) or computational (e.g., sequence similarity evidence), other types include author statement (with or without traceable reference) and curator inference.

In addition to the evidence, ECO describes the mechanism by which an assertion is made (e.g., manual by curator or electronic). the *"assertion method"* is defined as 'A means by which a statement is made about an entity'. For example, if an algorithm was used to assign a predicted function to protein without any curator judgment, ECO expresses that as automatic assertion. Similarly, if curator came up with the annotation after reading a result reported in a paper, ECO captures that a manual curation.

3 Databases

3.1 UniProt

UniProtKB (UniProt Knowledgebase) [10] is the worldwide primary database of protein sequence and highly annotated functional information. UniProtKB employs GO annotation, that associates GO terms (in MF, BP, and CC) with the UniProtKB records. This association is accompanied by the reference from which the evidence was derived, in addition to the evidence code that indicate the degree to which the annotation is supported. The evidence codes are commonly encoded as three letter "GO evidence codes". However, they are now being replaced by ECO (see Section 2.4) terms that provides the ability to capture more breadth and depth of evidence information than the traditional GO evidence codes.

Furthermore, each protein record contains a list of keywords that summarize the content of a UniProtKB entry and facilitate the search for proteins of interest. Keywords are controlled vocabulary with a hierarchical structure that are added during the manual annotation process. Generally, UniProtKB consists of two sections: Swiss-Prot and TrEMBL.

Swiss-Prot contains well-annotated, non-redundant proteins that have been manually inspected. The annotation includes protein and gene names, keywords assignment, function, sub-cellular

location, peer-reviewed references, secondary structure elements, and cross-references to other biological databases and information about their function. Most GO annotations in the Swiss-Prot are supported by ECO manual curation evidence terms.

TrEMBL contains protein sequences that are unrevised and automatically annotated. All of the GO terms are associated with ECO terms that are based automatic assertion. In addition, TrEMBL entries generally contain fewer keywords than Swiss-Prot entries, and the keywords are assigned automatically according to specific annotation rules.

As of September 2017, Swiss-Prot contains 557,713 sequence entries and TrEMBL contains 116,030,110 sequence entries.

3.2 Transporter Classification Database

The TCDB [11] uses a classification system approved by the International Union of Biochemistry and Molecular Biology (IUBMB) for membrane transport proteins; it is known as the transporter classification (TC) system. The TCDB is a curated database of accurate and experimentally characterized information collected from over 10,000 published references. As of August 2018, it contains more than 17,000 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 to 9 that corresponds to the transporter class (e.g., channels, carrier, pumps (active transport)), W refers to a transporter subclass, X is a number that refers to the transporter family, Y is a number that corresponds to transporter subfamily and Z refers to the substrate or range of substrates transported. 2 exemplifies a TCDB entry.

The 17,000 entries in TCDB are divided into 66 superfamilies, 1248 families, and 2554 subfamilies. The transporter class TC.9 contains all of the uncharacterized transporters with around 2630 entries. In total, there are 4016 sequences in TCDB that contains "unknown" on blank substrate identity.

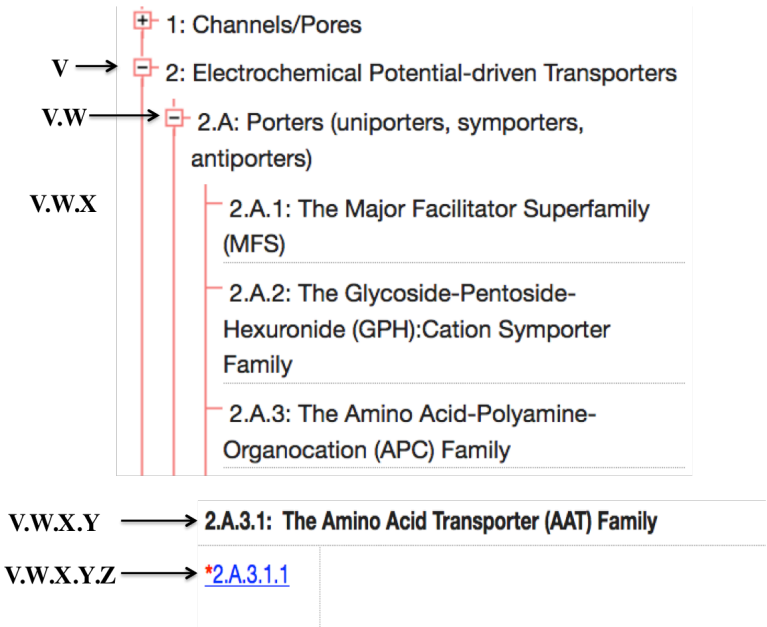

Figure 2: TCDB entry example

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.

4 Dataset I: membrane protein classes

This dataset includes membrane proteins sequences divided based on their biological roles: *enzymes, receptors, transporters*. It also contains membrane proteins that have other functions marked as *others*.

4.1 Materials and methods

we collected the data from the worldwide curated protein-sequence database, Swiss-Prot (Section 3.1), which we accessed in February of 2018. The membrane protein dataset contains four membrane functional classes: enzymes, receptors, transporters, and others.

The first set is the membrane proteins set S^{Mem} that is retrieved by searching as follows:

S^{Mem} = `locations:(location:membrane) reviewed:yes`

For the enzymes, we retrieved all sequences located in the membrane and annotated with EC numbers. The EC numbers for enzyme proteins are manually curated in the description of Swiss-Prot entries. The set of enzymes E_X was retrieved by searching as follows:

```
E_X = locations:(location:membrane) ec:X.-.-.- AND reviewed:yes
```

where $X \in \{1, 2, 3, 4, 5, 6\}$, the six enzyme reaction classes. The enzyme initial set E_0 is the combination of the six sets as:

$$E_0 = \bigcup_{x=1}^{6} E_x \tag{1}$$

For receptors, because they do not have a distinct classification like enzymes, we relied on Swiss-Prot curators keywords to retrieve the initial set. We chose a "Receptor [KW-0675]" keyword that falls into a molecular function keyword subcategory which indicates that the protein functions molecularly as a receptor. The initial set R_0 was obtain by searching as follows:

```
R_0 = locations:(location:membrane) keyword:"Receptor [KW-0675]"
AND reviewed:yes
```

For transporters, we search for proteins that have (GO:0005215 transporter activity) GO MF annotation. The GO MF is directly related to actual function of the protein rather than the general process that the it is involved in.

The initial set of transporters T_0 was obtained by searching as follows:

```
T_0 = locations:(location:membrane)
goa:("transporter activity [5215]") AND reviewed:yes
```

Finally, another initial set O_0 for other classes of membrane proteins was obtained by removing E_0, R_0 and T_0 from the membrane protein set S^{Mem} as follows:

$$O_0 = S^{Mem} \cap (E_0 \cup R_0 \cup T_0) \tag{2}$$

After the initial extraction of the membrane protein classes, we retrieved more information regarding each sequence from UniProtKB Swiss-Prot entries. The information includes keywords, organism species, subcellular locations, and GO MF annotations with the evidence codes using web scraping in R.

The data was screened to attain the best-quality dataset by adhering to the following criteria:

- **Step 1:**Protein sequences that have evidence "inferred from homology" for the existence of a protein were removed.
- **Step 2:** Protein sequences that are annotated with multiple functions (e.g., receptors and enzymes) were set aside for further examination.
- **Step 3:** Protein sequences that have no or merely computational evidence (IEA) of GO molecular function annotation were eliminated.
- **Step 4:** Protein sequences with more than 60% pairwise sequence identity were removed via a CD-HIT [12] program to avoid any homology bias.

The final sets are appointed after Step 4 for all the membrane classes —except in the case of transporters to which further filtering was applied —as is discussed in Section 5.

4.2 Results

The membrane proteins classes dataset and the details about the curation process of membrane proteins are presented in Fig 3. The final dataset contains total of 9063 membrane proteins sequences divided based on their functional role: 2890 enzymes, 1123 receptors, 2231 transporters, and 2819 belong to other functional classes.

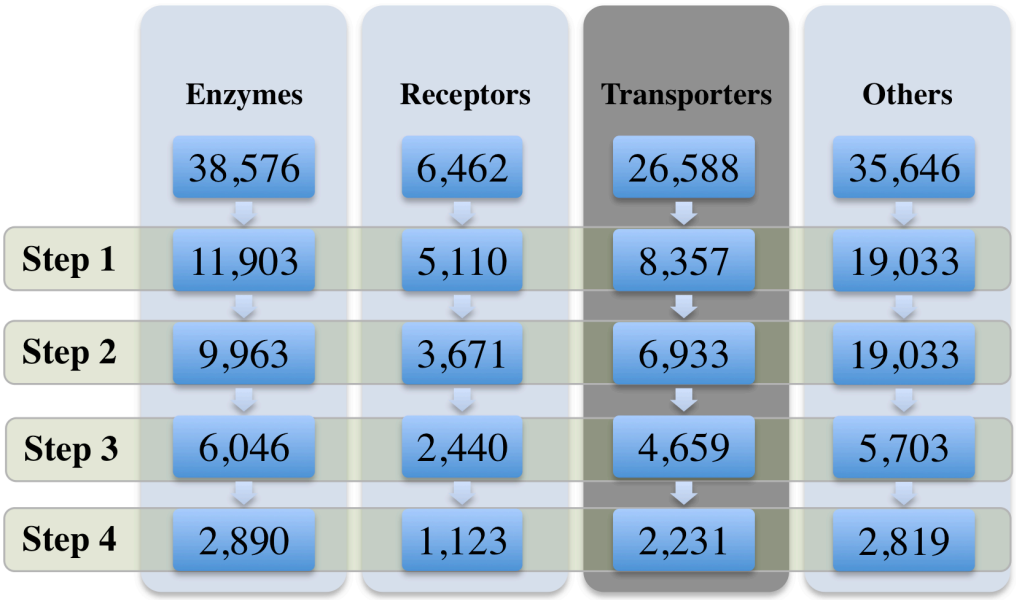

Figure 3: Membrane proteins curation process

This figure shows details of the curation process for membrane proteins. Step 1: Protein sequences that have evidence "inferred from homology for the existence of a protein were removed. Step 2: Protein sequences annotated with multiple functions (e.g., as receptors and enzymes) were set aside for further examination. Step 3: Protein sequences with no or merely computational evidence (IEA) for GO molecular function annotation were eliminated. Step 4: Protein sequences with more than 60% pairwise sequence identity were removed.

5 Dataset II: transporter substrate categories

This dataset contains transporter protein sequences divided based in their substrate categories. The latest gold-standard, substrate-specific, transport protein dataset was published by Mishra *et al.* [4] in 2014 . Their dataset consists of 900 transporters divided into seven major classes based on substrate specificity: 85 amino acid/oligopeptide transporters, 72 anion transporters, 296 cation transporters, 70 electron transporters, 85 protein/mRNA transporters, 72 sugar transporters, and 220 other transporters. However, due to the exponential growth of bioinformatical databases and protein annotation, we needed to update this dataset to include newly annotated proteins. While all of the available substrate prediction databases rely heavily on the manual curation, we adopted an ontology-based form of automatic substrate annotation. The automated approach revolves around GO MF annotation in Swiss-Prot entries, GO ontology, and CheBI ontology.

5.1 Materials and methods

The first decision we had to make was to choose the substrate categories with respect to ChEBI-IDs. We initially attempted to follow Saiers classification system [13] (See Table 2) by mapping each subcategory to its relevant ChEBI term, but we ran into multiple issues. First, the classification system simultaneously offers role and chemical classification. For example, category five (vitamins, cofactors, and their precursors) and some of the subcategories of category six belong to ChEBIs role ontology while the rest of categories belong to its chemical ontology. A single compound could have both if it includes *"is a"* and *"has role"* relations in its ontology.

Therefore, all of the compounds have chemical classifications and some also have role classifications. We would often run into the issue of multiple classification for a single substrate. For example, Glycine (CHEBI:15428), which is an amino acid, is classified under Sair classification as follows:

```
3.A. Amino acids and conjugates;5.D signaling molecules;
6.B. Specific drugs drug
```

Since we are most interested in the chemical composition of the transported substrates, and as

all of the substrate prediction methods (see Section **??**) predict the chemical classification, we have opted to consider the chemical categories.

Category and substrate type	**Subcategories**
1.Inorganic molecules	A. Nonselective B. Water C. Cations D. Anions E. Others
2.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
3.Amino acids and their derivatives	A. Amino acids and conjugates B. Amines, amides, and polyamines C. Peptides D. Other related organocations E. Others
4.Bases and their derivatives	A. (Nucleo)bases B. Nucleosides C. Nucleotides D. Other nucleobase derivatives E. Others
5.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
6.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
7.Miscellaneous compounds	

Table 2: Classification of transport system substrates based on biological significance

The second issue concerns the fact that the grouping of Saier's classification system categories and ChEBI ontology is not consistent. For example, there is no corresponding ChEBI term to 2.A sugars polyols and their derivatives subcategory but rather two different terms polyol (CHEBI:26191) and monosaccharide (CHEBI:63367). The closest common ancestor between these two termsis organic molecular entity (CHEBI:50860).

Similarly, monosaccharide (CHEBI:63367) and carbohydrate (CHEBI:16646) share ancestor "carbohydrates and carbohydrate derivatives" (CHEBI:78616) in the ChEBI ontology but are not in the same major category in Saier's classification system. Since we rely on the ChEBI ontology in our automatic substrate assignment, we modified Saiers classification system to be consistent with the ChEBI ontology. Fig 4 depicts Saiers classification system categories with respect to ChEBI ontology, where the edges represent *"is a"* relation. Table 3 groups the categories according the relevant closest ancestor in agreement with ChEBI ontology.

By using the categories in Table 3, we implemented an ontology-based system (see Section 5.1) that assigns substrate categories based on transporter protein GO annotations.

Category and substrate type	Subcategories	ChEBI-ID
1.Inorganic molecules	A. Nonselective	CHEBI:36914 CHEBI:24431
	B. Water	CHEBI:15377
	C. Cations	CHEBI:36915
	D. Anions	CHEBI:24834
2.Organic ion	A.Organic cation	CHEBI:25697
	B.Organic anion	CHEBI:25696
3.Carbohydrates and derivatives	A. Monosaccharide and derivatives	CHEBI:35381 CHEBI:63367
	B. Oligosaccharide and derivatives	CHEBI:50699 CHEBI:63563
	C. Polysaccharide and derivatives	CHEBI:18154 CHEBI:65212
4.Carboxylic acid	A. Monocarboxylic acid	CHEBI:25384
	B. Tricarboxylic acid	CHEBI:27093
	C. Dicarboxylic acid	CHEBI:35692
5.Organonitrogen compound	A. Amino acid	CHEBI:33709
	B. Amino acid derivative	CHEBI:83821
	C. Peptide	CHEBI:16670
	D. Amine	CHEBI:32952
	E. Polyamine	CHEBI:88061
	F. Protein	CHEBI:36080
	G. Other Organic amino	CHEBI:50047
6.Organic heterocyclic compound	A. Nucleobase	CHEBI:18282
	B. Nucleoside	CHEBI:33838
	C. Nucleic acid	CHEBI:33696
	D. Nucleotide	CHEBI:36976
7.Miscellaneous	A. Polyol	CHEBI:26191
	B. Organic phosphate	CHEBI:25703
	C. Amide	CHEBI:32988
	D. Other Organic	CHEBI:50860

Table 3: Classification of transport system substrates according on ChEBI ontology

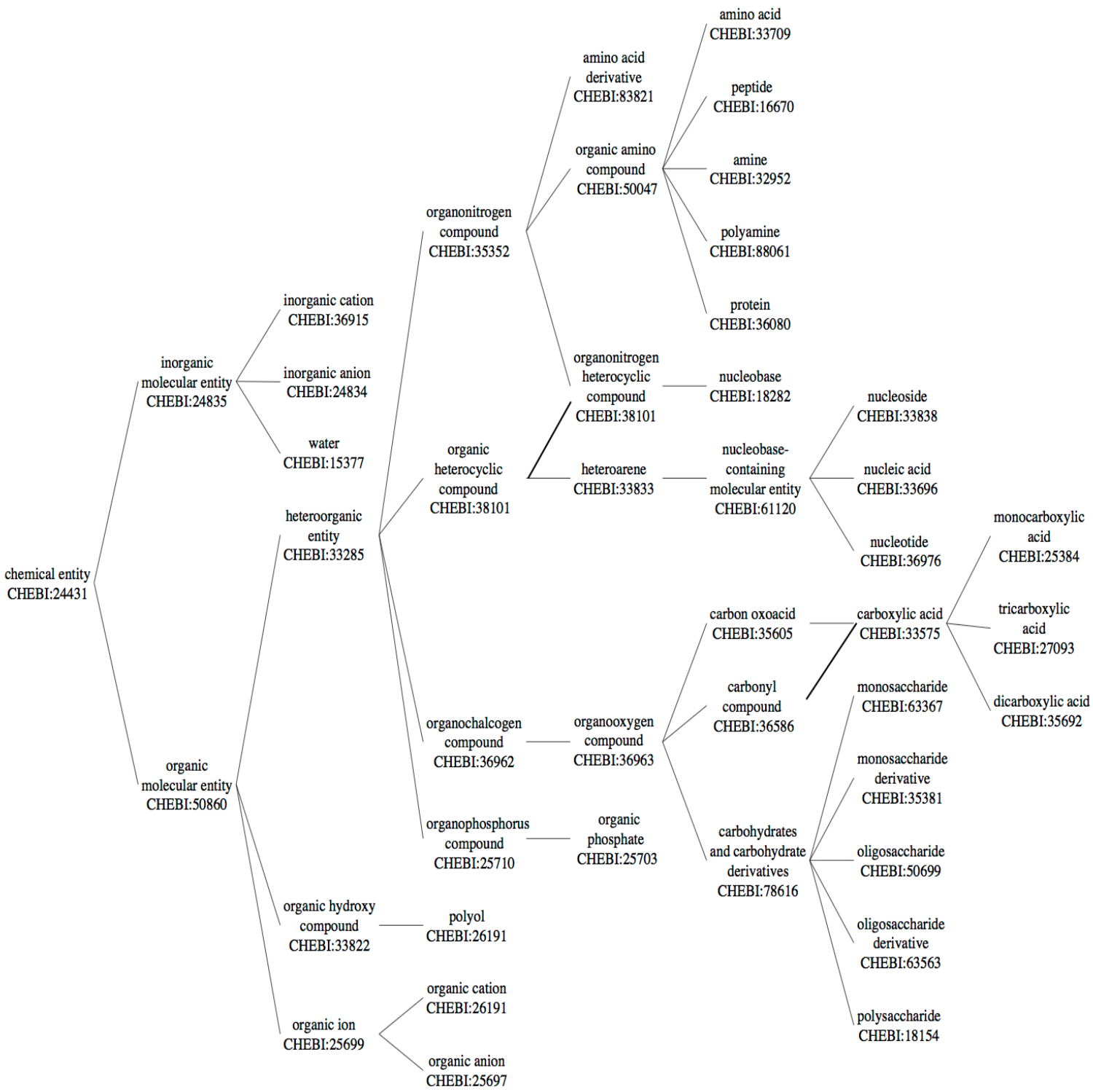

Figure 4: Simplified view of ChEBI ontology terms

This figure shows a simplified view of the categories in Saier's classification system with respect to ChEBI ontology. This tree, is organized from left to right where the edges represent *"is a"* relations in ChEBI ontology, some edges were omitted to simplify the view. Each node contains the ChEBI term and the relevant ChEBI-ID. The leafs are the categories.

Ontology-based automated substrate annotation

Here, we build automated, ontology-based substrate category annotation system. An overview of the system is presented in Fig 5. This system take the transporter protein uniprotID as an input, and outputs its substrate category as specified in Table 3. This section presents the implementation and details conducted to achieve the automation.

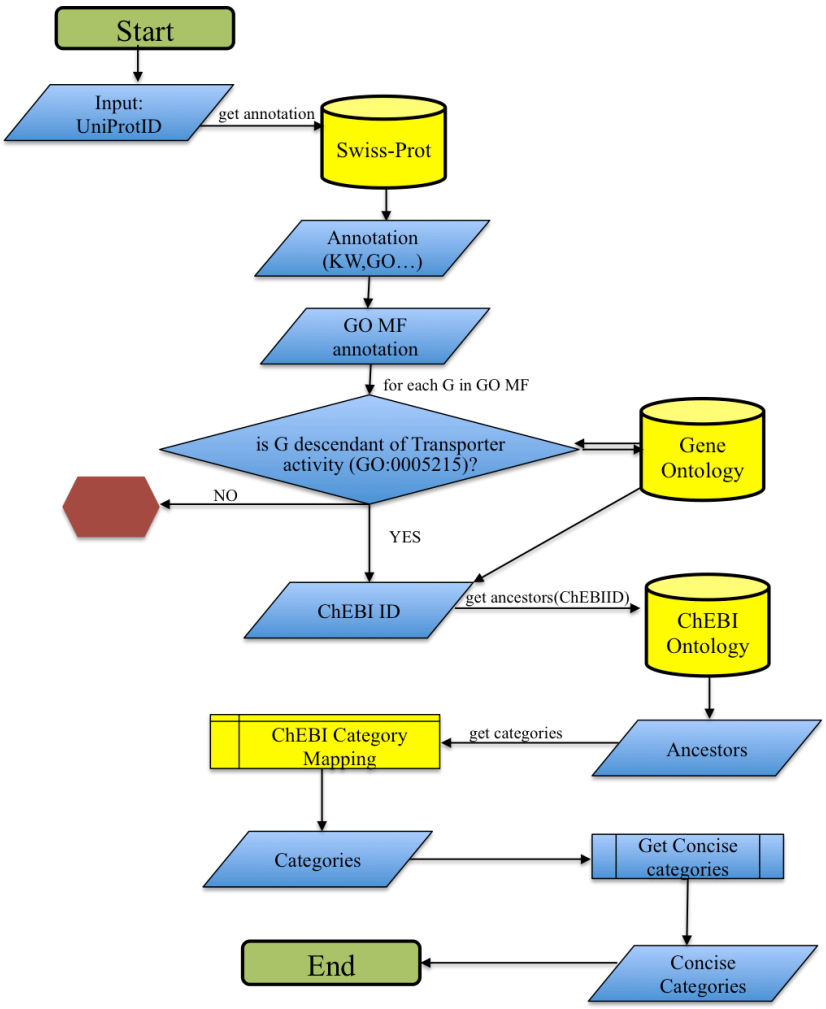

Figure 5: Ontology-based substrate category mapping overview

The system uses the transporter protein Uniprot-ID to retrieve its GO-MF annotation from the Swiss-Prot database. Then, for each GO-MF term in the protein annotation, it checks whether that term is a descendant of (transporter activity GO:0005215) in GO ontology. If yes, it gets its corresponding ChEBI-ID. Next, it uses ChEBI ontology to retrieve all of the ancestor terms of that ChEBI-ID. Then, using the categories with their relevant ChEBI-IDs in Table 3 it gets the initial category list. This list gets filtered to keep the most concise category, which is the output.

Algorithm 1presents the a pre-processing step, in which we retrieve all of the descendants of (GO:0005215 transporter activity) in Gene ontology, and get their corresponding ChEBI mapping. CheBI mapping is only available in *go-plus.owl* which is downloaded in August 2018, from `http://snapshot.geneontology.org/ontology/extensions/go-plus.owl`

Algorithm 1 Pre-process GO Chebi Mapping

Require: go-plus.owl file as GO ontology file
Ensure: mappingList$\langle term, list\langle ChEBI_ID\rangle\rangle$ that contains the mapping between transporter GO terms and ChEBI

```
function getTransporterGOChebi
    geneOntology ← getOntology("go-plus.owl")
    transporterRelatedGO ←getDescendants(geneOntology,"GO:0005215")
    mappingList← empty list⟨term, list⟨ChEBI_ID⟩⟩
    for term ∈ transporterRelatedGO do
        allChEBI ← getChebiRelation(term,GeneOntology)
        mappingList ← add(term, allChEBI − IDs)
    end for
    return mappingList
end function
```

As a result, *mappingList* contains 1080 descendants terms of (GO:0005215 transporter activity), 775 of them, have ChEBI mapping. Then, Algorithm 2 assigns the categories in Table 3 to the each term by traversing ChEBI ontology. We use *chebi_lite.obo* file that contains Chebi ids, names, subsets and relationships to traverse the ontology, downloaded in August 2018, from: `ftp://ftp.ebi.ac.uk/pub/databases/chebi/ontology/chebi_lite.obo`
Algorithm 2 calls Algorithm 3 to get the most concise categories. for example, the initial category mapping could be:

```
1.A Nonselective, 4.A amino acid, 4.G Other organic amino,
6.E Other organic
```

and the concise category is:

```
4.A amino acid
```

Algorithm 2 will result in creating *categorylist* which contains descendants terms of (GO:0005215 transporter activity) and their corresponding substrate categories as in Table 3.
Next, Algorithm 4 deals with the 2,231 transporter proteins in the membrane protein dataset (Section 4) and assign substrate categories to them. The categories are found by examining

Algorithm 2 Traverse ChEBI ontology

Require: chebi_lite.obo file as ChEBI ontology file
Require: mappinglist$\langle term, list\langle ChEBI_ID\rangle\rangle$ that contains transporter GO terms and CheBI mapping
Require: CategoryChebi$\langle name, ChEBI_ID\rangle$ that contains the ChEBI_IDs of the substrate categories
Ensure: substrate category mapping to each term in mappinglist

```
function TRAVERSEONTOLOGY(mappinglist⟨term, ChEBI⟩)
    ChEBIOntology ← getOntology("chebi_lite.obo")
    categorylist⟨term, list⟨name, ChEBI_ID⟩⟩ ← empty list
    for term ∈ mappinglist⟨term, list⟨ChEBI_ID⟩ do
        conciseCategories⟨ChEBI_ID⟩ ← empty list
        for  ChEBI_ID ∈ term do
            ancestors ← getAncestors(ChEBIOntology,ChEBI_ID)
            initialCategories ← CategoryChebi⟨name, ChEBI_ID⟩ where
            ChEBI_ID ∈ ancestors)
            conciseCategories ←add(getConciseCategory(initialCategories))
        end for
        categorylist ← add(term,unique(conciseCategories))
    end for
    return categorylist⟨term, list⟨name, ChEBI_ID⟩⟩
end function
```

the GO MF annotation that are descendants terms of (GO:0005215 transporter activity) of each protein and assigning their corresponding categories as determined by Algorithm 2. Furthermore, we assign evidence of each category as determined by GO annotation in Swiss-Prot for that sequence, this can enable us to determine the level of confidence in the substrate assignment.

For example Uniport **Q0WMZ5** has the following GO MF annotation:

```
GO:0015171 amino acid transmembrane transporter activity;IMP:TAIR,
GO:0042803 protein homodimerization activity; ISS:UniProtKB,
GO:0015288 porin activity; IEA:UniProtKB-KW.
```

only the first (GO:0015171) is descendant of (GO:0005215 transporter activity), its corresponding category is 4.A amino acid, The annotation evidence for that term is experimental (see Section 3.1) with code Inferred from Mutant Phenotype (IMP). Because the evidence is experimental, we have high confidence in this assignment. This protein will get the following substrate category:

```
4.A amino acid;IMP:TAIR
```

Algorithm 3 Get concise category

Require: initialCategoryMapping$\langle name, ChEBI_ID\rangle$ of initial category mapping
Require: ChEBIOntology as ChEBI ontology objecr
Require: mappinglist$\langle term, ChEBI_ID\rangle$ that contains transporter GO terms and CheBI mapping
Ensure: concise category mapping to the initial category

```
function getConciseCategory(initialCategoryMapping⟨name, ChEBI_ID⟩ )
    categoryAncestors← empty list⟨ChEBI_ID, list⟨ChEBI_ID⟩⟩
    for ChEBI_ID ∈ initialCategoryMapping⟨name, ChEBI_ID⟩ do
        categroyAncestor← getAncestors(ChEBIOntology,ChEBI_ID)
        categoryAncestors← add( ChEBI_ID,categroyAncestor)
    end for
    categoryPairs← getAllPairs(initialCategoryMapping⟨ChEBI_ID⟩ )
    unconciseCategories ← list⟨ChEBI_ID⟩
    for pair ∈ categoryPairs do
        firstPairAncestors ← categoryAncestors(pair[1])
        secondPairAncestors ← categoryAncestors(pair[2])
        if pair[2]∈ firstPairAncestors then
            unconciseCategories← add(pair[2])
        else if pair[1]∈ secondPairAncestors then
            unconciseCategories← add(pair[1])
        end if
    end for
    return initialCategoryMapping⟨name, ChEBI_ID⟩ where
    ChEBI_ID ∉ unconciseCategories⟨ChEBI_ID⟩
end function
```

Algorithm 4 Automatic substrate assignment to transporter sequences

Require: info$\langle uniprotID, list\langle GOMF\rangle, ...\rangle$ that contains the transporter GOMF annotations
Require: categorylist$\langle term, list\langle name, ChEBI_ID\rangle\rangle$ that contains descendants GO terms of (GO:0005215 transporter activity) and their corresponding substrate categories
Ensure: substrate category mapping to each transporter sequence

```
function assignSubstrate
    uniprotCategoriesMapping ⟨uniprotID, list⟨name, evidance⟩⟩ ← empty list
    for tuple ∈ info⟨uniprotID, list⟨GOMF⟩, ...⟩ do
        sequenceCategories⟨name, evidance⟩ ← empty list
        for GOMF ∈ tuple do
            if GOMF[term] ∈ categorylist then
                sequenceCategories ← add (getName(categorylist,term),GOMF[evidance])
            end if
        end for
        uniprotCategoriesMapping ← add(uniprotID ∈ tuple,removeDuplicate(sequenceCategories))
    end for
    return uniprotCategoriesMapping⟨uniprotID, list⟨name, evidance⟩⟩
end function
```

5.2 Results

As a result, the number of proteins in each category is presented in Table 4.

Category and substrate type	**Subcategories**	**Number of Sequences**	**Total**
1.Inorganic molecules	A. Nonselective	26	758
	B. Water	26	
	C. Cations	603	
	D. Anions	103	
2.Organic ion	A.Organic cation	13	120
	B.Organic anion	107	
3.Carbohydrates and derivatives	A. Monosaccharide and derivatives	126	151
	B. Oligosaccharide and derivatives	21	
	C. Polysaccharide and derivatives	4	
4.Carboxylic acid	A. Monocarboxylic acid	28	33
	B. Tricarboxylic acid	4	
	C. Dicarboxylic acid	1	
5.Organonitrogen compound	A. Amino acid	148	322
	B. Amino acid derivative	14	
	C. Peptide	27	
	D. Amine	4	
	E. Polyamine	11	
	F. Protein	113	
	G. Other Organic amino	5	
6.Organic heterocyclic compound	A. Nucleobase	18	61
	B. Nucleoside	16	
	C. Nucleic acid	3	
	D. Nucleotide	24	
7.Miscellaneous	A. Polyol	4	101
	B. Organic phosphate	23	
	C. Amide	7	
	D. Other Organic	67	

Table 4: Number of sequences in each substrate category

In Table 5 we compare the results of the dataset created by our automated ontology-based substrate category mapping with Mishra's *et al.* [4] TrSSP dataset that was mapped manually. Out of the 900 protein sequences in TrSSP dataset, 565 are present in our transporter dataset —the rest were filtered in the curation process (see Section 4). While direct comparison is difficult because Mishra has only seven substrates categories, one of them is "others". We grouped our organic, inorganic cations into Mishra's cation transporter class, and our organic, inorganic anion into TrSSP anion transporter class, and our amino acid, amino acid derivatives into amino acid transporter class, then our monosaccharide and derivatives class into their sugar class, our protein class has direct protein class. As a result, we found that 459 proteins have the same mapping, Then, 50 of the proteins in TrSSP dataset do not have sufficient GO terms, thus no mapping in our case. Finally, 48 proteins (about 8%) do not have the same category.

		Ontology-based		
Transporter class	**TrSSP dataset**	**Agreement**	**No mapping**	**Disagreement**
Amino acid	65	63	0	2
Anion	53	45	1	7
Cation	193	180	5	12
Protein	40	30	10	1
Sugar	51	41	4	6
Electron	2	0	0	2
Other	149	100	30	19
Total	565	459 (82.55%)	50 (8.99%)	49 (8.81%)

Table 5: Comparison between Mishra's TrSSP dataset and our ontology-based dataset

This table shows the number of transporter classes Mishra's TrSSP dataset that are also in our ontology-based dataset. Because the transporter categories are not the same between the two datasets. we mapped our categories to TrSSP dataset categories as follows: amino acid, amino acid derivatives (5.A,5.B) into amino acid. Our organic, inorganic anion (1.D, 2.B) into anion transporter class. Organic, inorganic cations (1.C, 2.A) into Mishra's cation class. Monosaccharide, oligosaccharide and derivatives classes (3.A,3.B) into sugar class. Our protein (5.E) to their class. The rest of our categories are mapped to others. **Agreement** indicates that ontology-based perdition dataset have the same category as TrSSP dataset category. **No mapping** indicates that there is no corresponding category. **Disagreement** indicates that ontology-based perdition dataset and TrSSP dataset have different categories.

5.3 Discussion

We have 2,231 transporter proteins in the membrane protein dataset (Section 4). 1,546 of them have clear substrate annotation, 379 have no ChEBI mapping, and 306 have multi-class annotations. In Table 4, we see the number of proteins assigned to each category. The category with the largest number of proteins is 1.C (*inorganic cations*), with 603 transporters. This is not surprising, as ion channel transporters comprise a large class which transport ions such as potassium, sodium, and calcium. This is also evident in other substrate-specificity datasets, such as the dataset used in Mishra's *et al.* [4] TrSSP method , in which 296/900 proteins belong to the cation category. The category with the second highest number is 5.A (*amino acids*) with 148 proteins, followed by 4.A (*monosaccharide and derivatives, also known as sugars*) with 126 sequences. Then 2.B (*proteins*), 5.F (*organic anions*), and 1.D (*inorganic anions*) with 113, 107, and 103 proteins, respectively.

In the rest of the categories, there are significantly fewer proteins. Though we did not build a dataset for exact substrates (e.g., benzoate), and though we grouped them into larger categories (e.g., monocarboxylic acids), we still acquired a quite small number of proteins. This further highlights the fact that membrane protein are still not very well characterized compared to other types of proteins. Thus, it may not be feasible at this time to proceed to the level of exact substrate prediction. Thus, a major decision we need to make concerns how to deal with the categories with the fewest proteins. When predicting the substrate specificity of transporters, we need to determine how specific we can get without sacrificing the overall performance of the system. Some of the substrate-specificity predictors (see Section ??) have opted to group the small categories into *"others"* [1] [4], while others only predict the significant categories [5] [3] [2].

When we compare the categories in the dataset created by our automated ontology-based method to those of TrSSP dataset, we find that there are disagreements between the GO MF annotation and the manually assigned transporter class in TrSSP dataset. This could be due to the simple fact that protein annotations are created and updated regularly, or that curators have different interpretations of the literature. Nevertheless, our ontology-based, automated substrate annotation has an edge over all of the other substrate predictors datasets that were built manually because it exploits the already manually annotated GO MF of Swiss-Prot entries and uses them in automated manner. Not only this makes it more standardized and consistent, but also easily adjusted to the ever-increasing size of biological databases.

6 Conclusion

We built two datasets. The first dataset contains the four functional classes of membrane proteins: enzymes, receptors, transporters, and others. The second dataset contains substrate-specific transporter proteins. Because membrane proteins are not well characterized, and because there is not enough data to predict the exact substrates (e.g., arginine), all transporter substrate prediction occurs at the level of substrate category or class (e.g., amino acid). Here, we assigned the substrate categories to transporters by using our ontology-based automated substrate annotation system, which can easily be adjusted to accumulate more or fewer categories. This system can adapt to the exponential growth of the biological databases and assign the substrate automatically. Not only does this automation relieve us of the huge burden of manual curation used to build all of existent transporters substrate databases, it is is consistent with other ontologies and is reproducible.

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, and C. Steinbeck, "ChEBI in 2016: Improved services and an expanding collection of metabolites," *Nucleic Acids Research*, vol. 44, no. D1, pp. D1214–D1219, 2015.

[7] D. P. Hill, N. Adams, M. Bada, C. Batchelor, T. Z. Berardini, H. Dietze, H. J. Drabkin, M. Ennis, R. E. Foulger, M. A. Harris *et al.*, "Dovetailing biology and chemistry: integrating the gene ontology with the chebi chemical ontology," *BMC genomics*, vol. 14, no. 1, p. 513, 2013.

[8] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, p. 25, 2000.

[9] M. C. Chibucos, C. J. Mungall, R. Balakrishnan, K. R. Christie, R. P. Huntley, O. White, J. A. Blake, S. E. Lewis, and M. Giglio, "Standardized description of scientific evidence using the Evidence Ontology (ECO)," *Database*, vol. 2014, 2014.

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

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

[13] 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.