Prediction of transporter proteins

Jie Xiong

Helsinki October 14, 2010
Seminar paper
Master’s Degree Programme in Bioinformatics
UNIVERSITY OF HELSINKI
Department of Computer Science
The motivation of using machine-learning approaches to classify transporter proteins according to their function is stated. Different approaches are introduced, compared and discussed. The emphasis is laid on the representation of the input vectors, as well as selecting of training data sets.
## Contents

1 Introduction ........................................... 1

2 Algorithms ......................................... 1

3 Representation of feature .......................... 2
   3.1 A weighting strategy for K-NN ................. 2
   3.2 Vector construction for SVM ..................... 4
   3.3 Position-specific scoring matrices and biochemical properties ....... 6

4 Training data ...................................... 6

5 Discussion and conclusion ......................... 8

References ....................................... 10
1 Introduction

Membrane proteins perform vital biological functions with high diversity, such as importing and exporting ions and molecules across membranes, attaching to small molecules, or identifying the immune system. Transporters are a major class of membrane proteins, and through them cells are able to deliver essential nutrients to maintain ion concentration, eject waste products to prevent the accumulation of toxins and transport signal molecules to assist intercellular communications [Yan03]. Various experimental Bio-techniques have been designed to discover transporters as well as their mechanisms [Yan03]. For instance, patch clamp techniques have been applied to recognize transported substrates and their mechanism of ionic channels in excitable membranes [SN84]. However, those experimental techniques are not only costly and time-consuming, but also require sophisticated skills. As a result, computational methods are designed to select candidates for maximizing the experimental study. In this seminar report, we will first show different approaches for transporter prediction by introducing the algorithms, representations and the selection of training data sets and finally have a discussion among them with the motivation, performance and restriction.

2 Algorithms

The central issues of the literatures included in this report are not about developing or modifying new machine learning algorithms. Most applied machine-learning approaches were carried out with existing packages [LDZ08][GY08][OCG10] and no remarkable breakthrough of algorithms were reported. However, different machine learning algorithms may have different requirements in the design of representations and training data sets. Therefore, here we briefly introduce several machine-learning algorithms which were applied in the literatures included in the seminar report.

• Nearest neighbour
In the nearest neighbour classification, the unknown protein is assigned the label(class, family, etc.) of a previously described transporter in the training data set, which is most similar to the unknown protein [LDZ08]. An extension of the nearest neighbour classification is the k nearest neighbour classification(K-NN). In K-NN, the unknown protein is assigned the label most frequently represented among the K training transporters, which are most similar to the unknown protein. In other word, the decision is made by first selecting the k nearest training transporter and then making a majority voting [L+]. Therefore, the most important part of a nearest neighbour algorithm is the definition of similarity measurement.

• Support vector machines
In a support vector machines (SVM), protein sequences are processed into numerical vectors in a high dimensional space, where proteins in a family are separated from those outside the family by a hyperplane \((LHC^+06)\). The goal in training SVM is to find the hyperplane with the largest margin, which is the positive distance from the hyperplane to the nearest data. It is expected that a larger margin provide a better generalization of the classifier. The support vectors are the vectors of processed training sequences that are equally close to the hyperplane, which defines the optimal hyperplane and are the most difficult patterns to classify. That is, the support vectors carries the most important information of the classifier \((L^+)\).

**Remarks:** we introduce these two particular algorithms, simply because some approaches included in the report are customized for the algorithms to some extent \([LDZ08] [LHC^+06]\) and in fact, many other machine-learning algorithms can be applied in the prediction of transporters such as radial basis function networks \([OCG10]\). In the following section, we will show how Li et al. \([LDZ08]\) designed representation for their K-NN framework and Lin et al. \([LHC^+06]\) selected training data set for the SVM.

### 3 Representation of feature

In this report, the representation of the biological problem is the issue of most interest. In this section, we introduce several different methods how biological information can be encoded into numerical input vectors. Sequence homology searches, motif searches as well as transmembrane segments analysis were employed by Li et al. \([LDZ08]\) under a weighting strategy, which is customized for their K-NN framework. Amino acid composition and biochemical properties are adopted by both Lin et al. \([LHC^+06]\) and Gromiha et al. \([GY08]\| [OCG10]\), although they have different expression on the latter term. Moreover, Gromiha \([OCG10]\) took position specific scoring matrix \((PSSM)\) profiles into account to achieve a higher accuracy.

#### 3.1 A weighting strategy for K-NN

To define the similarity measurement for the K-NN framework, Li et al. \([LDZ08]\) applied a weighting strategy on homology searches and motif searches and adopted transmembrane segment analysis to handle the negative information.

- **Homology searches**
  Query protein sequences are searched for similarity against the database of previous identified transporter sequences and characterized as putative transporters if homologous sequence are found. BLAST is a widely used tool in the homology search \([A^+90]\). Although sequence-homology-based methods are simple, straight forward and may identify many real targets accurately, large number of false positives will be reported because diverse transport functions
may be evolved out of homologous sequences, especially paralogs [Doo81] and extensive manual annotation are still required after homology-search [LDZ08].

- **Motif searches**
  The idea of motif-based approaches are modelling motifs with tools like HMM for the protein families and query proteins are characterized as putative transporters using the motif model. However, these methods require a number of transporters for constructing the models and may fail by the low levels of conservation amongst transporter families. Although several databases provided some motifs about certain transporter families, dedicated and comprehensive information of transporter motifs is still limited. Therefore, these methods often miss a large percentage of putative targets [LDZ08].

- **Transmembrane segments (TMS) analysis**
  TMS methods are based on the number of transmembrane segments [LDZ08]. These methods are simple but often lead to significantly high false positive rates, because the transmembrane segments taken into account do not generally indicate transporter functions. Therefore, TMS-based methods alone are insufficient to predict transporter protein [LDZ08].

In the work of Li et al [LDZ08], the similarity measurement is a weighted distance of BLAST search score between an query protein and transporter protein in the training dataset and the motif HMM search score of the query protein against the family of the compared classified transporter. As a result, the query protein is classified into a transporter family that contained both members sharing highly sequence similarity with the query and members that contained motifs matched in the query protein [LDZ08].

However, in the weighting process, two practical issues must be taken into account. First, for some transporters, it is impossible to apply the weighting because of

- The family of the transporters in the training dataset may contain too few members to construct the motif models. Therefore, motif scores of query protein to this motif is unavailable.

- The transporters in the training dataset may be too dissimilar to the query protein, therefore searching tools like BLAST will not report them. As a result, sequence homologous values of the query protein against these transporters are not available.

Therefore, in the weighing strategy, for those query proteins which only get hits by a single search method, their scores of single measurements should be comparable with the ones with weighted measurements. Here, same measurements in the motif scoring (HMM) and sequence homology scoring (BLAST) are applied, namely e-value scores, and the square root of their product are assigned as the weighted score, if both measurement are available.
The second issue need to be taken into account is the negative training dataset, i.e. the training dataset for non-transporters. Usually, transporter databased only contain transport proteins without information about non-transport proteins, such as other membrane proteins without transport functions. Without the negative training data, traditional classification methods may not work with the query proteins which are actually non-transporter. Therefore, putative transmembrane segment(TMS) information are generated and introduced in order to circumvent this problem to some extent. Depending on the specific transport requirements and duplication processes that occurred during evolution, the numbers of transmembrane segments in most transporter families tended to vary slightly. Therefore, a minimal requirement for family membership is set as that 'the number of transmembrane segments for an query protein must be within 1 standard deviation of the mean number of transmembrane segments in that family for the candidate to be included in the family.' As non-transporters have either no TMS or the numbers of TMS outside of the normal distributions of many transporter families, this simple filter will eliminate most non-transporters while retaining most transporters for further analysis. Details of the weighting are shown in equation(1) ,respectively [LDZ08].

\[
\begin{align*}
  s(p, q_{ij}) &= \begin{cases} 
    \sqrt{\text{blast}(p, q_{ij}) \times \text{hmm}(p, MDL_i)} & |F_i| \geq k \& \text{tms}(p) \geq |\pi_i - \sigma_i| \\
    \text{hmm}(p, MDL_i) & |F_i| \geq k \& \text{tms}(p) \geq |\pi_i - \sigma_i| \\
    \text{blast}(p, q_{ij}) & |F_i| < k \& \text{blast}(p, q_n) \leq \tau
  \end{cases} 
\end{align*}
\]

Equation (1) define the similarity score between a training transporter \( q_{ij} \) (the \( j \)th transporter in Family \( F_i \)) and the query protein \( p \) in three cases. In the first two cases, the compared transporter family \( F_i \) has at least \( k \) members (i.e. available in HMM score) and the TMS of query protein \( p \) are checked against the family. If the tms conflicted, the query protein will be discarded. If its blast score is in the threshold \( \tau \), then the weighted score will be used. Otherwise, it will go to the second case, where the HMM score of \( p \) on the family will substitute all the members of the family. If the size of family \( F_i \) is smaller than the threshold \( k \), it will go to the last case, where only blast score is measured. The query protein will be discarded if the blast score is beyond threshold. In the formula, \( MDL_i \) is denoted as the HMM model of the family \( F_i \) if it is available, \( \pi_i \) and \( \sigma_i \) are the expected value and standard deviation of TMS number of \( F_i \) respectively. [LDZ08]

Remark: Equation(1) have slight differences with the formulas from original literature for the reason that notation of the original formulas is confusing.

### 3.2 Vector construction for SVM

Construction of the feature vector of each protein for SVM in Lin et al [LHC+06]is based on amino acid composition and several physicochemical properties including
hydrophobicity, normalized van der Waals volume, polarity, charge, surface tension, secondary structure and solvent accessibility [LHC+06].

- The amino acid composition (AAC) is adopted by both Lin et al [LHC+06] and Gromiha et al [GY08][OCG10]. It can be computed with the number of amino acid residues of each type and the total number of residues, which is defined as:

\[ \text{Comp}(i) = \frac{\sum n_i}{N} \]  

(2)

where \( i \) stands for one of the 20 amino acid residues, \( n_i \) is the number of residues of each type and \( N \) is the total number of residues. The summation is through all the residues in all the considered proteins [GY08]. For a single protein, the 20 elements of amino acid composition are the number of occurrences of 20 different amino acid normalized with total number of residues in a protein [OCG10].

Similarly, compositions of amino acid pairs (DPC) can be defined as a 400 vector whose value are the number of occurrences of 400 different amino acid pairs normalized with total number of residues in a protein [OCG10].

- For each of the other properties of one protein, amino acid residues are divided into three groups according the property. For instance, for van der Waals volume, amino acid residues can be divided into three groups, according to their value of van der Waals volume, that is, 0 – 2.78 (GASCTPD), 2.95 – 4.0 (NVEQIL), and 4.43 – 8.08 (MHKFRYW). Three descriptors, composition (C), transition (T) and distribution (D), are used to describe the property of one protein sequence. C is the number of amino acids in a particular group of the property divided by the total number of amino acid residues in a protein sequence. T is the percent frequency of the amino acid residues in a particular group followed by the ones in a different group. D represents the normalized chain length of the first, 25%, 50%, 75%, and 100% of the amino acid residues of a particular group are located, respectively.

**Remarks:** The methods of computing amino acid composition of SVM was not given by their literature [LHC+06], and was referred to the references. However, there was no expected description in the references they linked. Therefore, the definition of amino acid composition given here is the one using by Gromiha et al [GY08][OCG10], which is expected sharing the same idea. However, the data shown in [GY08] (i.e. Table 1 and Table 2) is conflict with the definition and context. One reason is that the two table of [GY08] are interchanged by mistake (answered by Gromiha’s e-mail). This explains the disagreement between the data and context around the data. Nevertheless, it will not significantly influence our understanding of this terminology. Although computing methods of the physicochemical properties were given in the literature [LHC+06], the description has mistakes and is very misleading. One possible reason is that the authors themselves were not interested in the computational processes. Their work could be considered as an extension of one previous publication by Cai et al. [CHJ+03]. That is, they did nothing remarkable on the algorithm,
as well as the vector construction methods. What they did is designing training data sets of transporters for the SVM-Prot (web-based support vector machine software for functional classification of a protein from its primary sequence) [CHJ+03] and testing the results, which we will show in the following sections.

3.3 Position-specific scoring matrices and biochemical properties

In an early article of Gromiha et al [GY08], amino acid composition were adopted as input vector for different machine learning frameworks and in a recent publication [OCG10], they developed a novel approach based on PSSM profiles and biochemical properties.

- Position specific scoring matrix (PSSM) profiles have wide application in various bioinformatics problems such as protein secondary structure prediction. Therefore, it can be introduced, in order to overcome the problem that amino acid residues can be mutated without changing the structure of a protein (i.e. proteins with different amino acid compositions may have similar structures). As shown in Figure 1, in the literature [OCG10], for each protein, a $20 \times 20$-dimensional input vector were generated by first summing up the same amino acid rows in PSSM profiles, which can be obtained by using PSI-BLAST against non-redundant (nr) protein database, and then normalized by the length of the sequence and scaled by $\frac{1}{1 + e^{-x}}$.

- Additional to the PSSM profiles, the biochemical properties of amino acid residues (Bio) are adopted by Gromiha et al [OCG10] as novel features to provide more information to the classifier. In the study [OCG10], totally 49 physical, chemical, energetic and conformational properties of amino acid residues were used. For each property, different values are taken by 20 amino acids. When we calculate the feature of a particular property of a protein, the sequence is divided into four parts, and the average values of amino acid residues of the four parts will be taken as the wanted feature. That is for a particular property of a protein, the feature is a $4 \times 1$ vector. Each of these 49 biochemical property features was added with the PSSM feature and tested one by one. Features who can improve the performance were remained.

4 Training data

Machine-learning approaches require large sets of training data. The transporter classification database, which employs a hierarchical, functional and phylogenetic
Figure 1: Method to get the input vector from the original PSSM profiles. [OCG10]

classification system, is used in the literatures included in this seminar report [LDZ08] [LHC+06] [GY08] [OCG10]. The transporters in TCDB are classified at different level (TC class, TC subclass, TC superfamily, TC family, and TC subfamily) [LHC+06]. In this system, transporters in the same family are expected to share similar transport functions. Therefore, if a query protein is classified into a family, its transport mechanism or pathway is predicted. Other system such as Pfam, whose classification system relies on sequence similarity, are not suitable to get such an inference on protein function [LDZ08].

Positive training data sets can be achieved by selecting transporters from TCDB according to different classification purpose. Li et al [LDZ08] downloaded 4155 transporters within 740 TC families from TCDB, then excluded the 256 families that contained only a single family member and remaining 3899 transporters within 484 TC families in the study. The motivation of exclusion was not mentioned in the original literature, which could be that query protein will hardly be assigned to the family with a single member in a K-NN framework with $K \geq 3$. For training the SVM of Lin et al [LHC+06], much larger dataset was required. In total, 13 TC subclasses with 14987 transporters and eight TC families with 2684 transporters were included in their work. In the earlier study of Gromiha et al [GY08], they select 510, 502 and 706 transporters from three major classes of transporters, channels/pores, electrochemical and active transporters respectively and finally form a dataset containing 1718 proteins. In the recent publication of them [OCG10], these numbers are reduced to 280, 170, and 243 respectively. The reason of smaller training dataset
could be the more complicated encoded representation vectors.

For negative training data sets, the task could be more tricky. In the K-NN framework of Li et al [LDZ08], they did not use non-transporter proteins of universal database to form a negative data set. Instead, they adopted TMS to distinguish the proteins outside the families. However, the framework may still suffer from lacking of negative information. In the SVM of Lin et al [LHC+06], for each TC subclass/family, the negative data set contains both non-transporters and transporters of other subclasses and families, and those non-transporters are selected from seed proteins of 7316 curated protein families of Pfam database that share no member with those TC subclasses/families. The motivation of doing this is trying to make the negative data evenly distributed among the space, covering as much negative information as possible. In the studies of Gromiha et al [GY08] [OCG10], query proteins could be classified by two steps, discrimination of transporters and non-transporters and discrimination of transporters based on classes/families. In the first step, the negative data set is formed by 18837 non-transporters obtained from UniProt database, and those non-transporter query will expected to be filtered out at this step. In the second step, negative data of a class/family is simply transporters from other classes/families.

Remarks : The description of the process choosing negative data sets of SVM of Lin et al [LHC+06] here is modified because the original description in the literature contains mistakes and is confusing [LHC+06].

5 Discussion and conclusion

So far, we have introduced several methods dealing with the transporter prediction, including the algorithms they used, the representation they designed as well as the training data set they selected.

The motivation of the K-NN framework by Li et al [LDZ08] is obvious. It can be considered as an extension of sequence homology search in order to cover as many transporter families as possible. Generally, compared to simply using BLAST, combining the three measurements (BLAST, HMM, TMS) will increase the validation rates but decrease the prediction rates(see Table 1 and 2 in [LDZ08]), because some query protein are discarded by the TMS filter. However, this this approach may suffer from lack of credible negative information in the training and be restricted with features presented in the database(i.e. some novel types of transporters will not be detected). Moreover, although this framework may overcome the problem of homology search that diverse transport functions may be evolved out of homologous sequences to some extent, it seems not to be able to prediction the dissimilar sequences sharing the same function. Never the less, this approach provide a novel strategy for the identification and classification of putative transport proteins based on the sequence similarity.[LDZ08]

It is reported that some transporters in different TC families have been found to
have very low sequence similarity to other family members. Therefore, sequence similarity based approach may fail classifying this kind of transporters. The motivation of Lin et al [LHC+06] using SVM is to avoid sequence similarity measurement to some extent. Somehow similar to K-NN framework of Li et al [LDZ08], this SVM [LHC+06] may also have problem related to the negative data. To a certain subclass/family, the size of negative data is often larger than the positive one, because the number of transporters in the class is significantly less that the non-members [LHC+06]. Therefore, there is an imbalance between the positive and negative training data set for each subclass and family. SVM tends to generate a hyperplane closer to the side with smaller number, leading to a low accuracy for assigning the subclass of transporters than that for the non-members. Another problem of this SVM approach [LHC+06] is that it requires large number of training sequences for each subclass/family, roughly 80-100. Therefore, those subclasses/families with small size will not be covered by the classifier.

In the studies of Gromiha et al [GY08] [OCG10], representation vectors were not specifically designed for any particular machine-algorithms and they could be adopted as input for any machine learning framework. In an early publication [GY08], they tested different machine learning methods with amino acid composition as input and neural network was reported having the highest accuracy among all the algorithms. In the recent work [OCG10], different combination of AAC, DPC, PSSM and Bio properties are applied as input vector into a radial basis functional networks framework (RBFN) and tested using with default setting. Using PSSM profiles with biochemical properties (PSSM + Bio) as input was reported having the best performance among all the combinations. This novel approach [OCG10] showed an significantly improvement in accuracy over previously reported methods with a relatively smaller training data and employed a better handling of negative data sets compared to other methods. However, it is worth to be mentioned, in the work of Gromiha et al [GY08] [OCG10], algorithms were not of interest and implementations were carried out by input those representation vectors to existing softwares such WEKA [GY08] and QuickRBF [OCG10] with default setting. Therefore, Gromiha et al [GY08] [OCG10] assessed their approaches only by numerical illustration without enough reasoning from analytical view. Moreover, Gromiha et al [GY08] [OCG10] did not mention any constraints of their approaches, which would be of interest for the further study.

Remarks:

- In the study of Gromiha et al[OCG10], not every combination of properties were tested. Therefore, it remains a question: 'Is there any other combinations getting better performance?'
- WEKA is a practical machine learning tool including several methods based on Bayes function, Neural Network, Logistic function, Support vector machine, Regression analysis, Nearest Neighbour, Meta learning, Decision tree and Rules [WF05].
- QuickRBF is a friendly and efficient software package for Radial Basis Function Networks [Ou05].
References


