Predicting Membrane Proteins Type Using Inter-domain Linker Knowledge

Nazar Zaki[1], **and Wassim El-Hajj**[2]
[1]Faculty of Info. Tech., UAE University, Al-Ain 17551, UAE.
nzaki@uaeu.ac.ae

[2] Electrical and Comp. Eng. Department, American University of Beirut, Beirut, Lebanon 1107-2020.
wassim.el-hajj@aub.edu.lb

Abstract - *Predicting membrane type is a crucial problem in computational biology. It is closely related to the biological function of the protein and its interaction process with other molecules in a biological system. The function of a membrane protein is closely correlated with the type it belongs to. In this paper we introduce DomMat, a novel method to predict membrane protein types. The method extracts functional domains by removing all the corresponding inter-domain linkers from the membrane protein sequence. A novel matching algorithm is then introduced to measure the sensitivity of the functional domains information to the membrane protein sequences of interest. Two protein sequences are expected to be related if they contain similar functional domain information. DomMat was tested in a high-quality benchmark dataset. The dataset consists of eight different membrane protein types collected from the Swiss-Prot database. The results obtained suggested that DomMat is comparable to the state-of-the–art methods and indeed a very useful method in identifying membrane protein types.*

Keywords: membrane protein; inter-domain linkers; support vector machines

1 Introduction

A membrane protein is a protein molecule that is attached to, or associated with the membrane of a cell or an organelle. Membrane proteins play key roles in controlling the processes of life. Given the importance of membrane proteins in various cellular processes, the roles they play in diseases and their potential as drug targets, it is imperative that the types of proteins be better studied [1]. The determination of function for new membrane proteins can be expedited significantly if we can find an effective scheme and algorithm to predict their types. The types of the membrane proteins are shown in Fig. 1. The function of a membrane protein is closely correlated with the type it belongs to. With the rapid increment of the number of protein sequences entering into public data banks; it would be both time-consuming and costly to rely on completely experimental work to predict membrane types. This is why the development of computational tools that are capable of predicting the types of membrane proteins is growing. Hence, computational approaches remain essential to assist in design and validation of the experimental studies. As a result, a vast set of impressive computational methods have been developed. However, most of the recent state-of –the-art computational methods (e.g. [2]-[6]) have one common drawback. They are either based on amino acid composition knowledge or considering all sequence-order effects by using pseudo amino acid composition. Predicting membrane protein types using amino acid composition is simple and provides information about the supplementary or complementary value of proteins. Nevertheless, a serious weakness is that using amino acid composition does not take into account the bioavailability of the amino acids and their structural relationships. Very little work is done to use protein structural knowledge to represent membrane protein sequences.

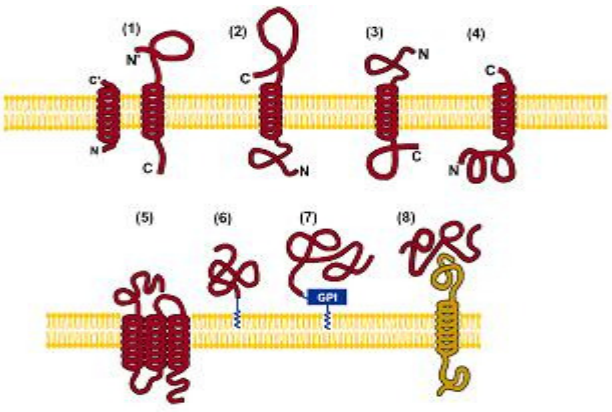

Figure 1: Schematic illustration to show the eight types of membrane proteins: (1) type I transmembrane, (2) type II, (3) type III, (4) type IV, (5) multipass transmembrane, (6) lipid-chain-anchored membrane, (7) GPI-anchored membrane, and (8) peripheral membrane [6].

In this paper, we utilize the structural knowledge as a way to represent the membrane protein sample. We introduced a novel technique to efficiently extract the protein

functional domain using inter-domain linkers regions as a way to incorporate structural knowledge. The identification of protein functional domains plays an important role in protein structure comparison. The comparison of membrane protein structures allows one to peer back farther into evolutionary time, based on the concept that a form or structure remains similar long after membrane sequence similarity has become undetectable [7]-[10]. Once the membrane protein is represented, a novel matching algorithm is introduced to measure the sensitivity of the extracted protein domains to the membrane protein sequence. A protein domain is a part of protein sequence and structure that can evolve, function, and exist independently of the rest of the protein chain. Each domain usually forms a compact three-dimensional structure and often can be independently stable and folded. Many proteins consist of several structural domains. One domain may appear in a variety of evolutionarily related membrane protein sequences. A powerful machine learning algorithm such as support vector machine (SVM) is then utilized to discriminate between eight membrane protein types.

2 Method

The algorithm for predicting membrane protein types based on functional **dom**ain **mat**ching (DomMat) consists of two major steps:

(1) *Feature extraction step:* In this step we represent each protein sequence by its sensitivity to a set of short amino acids sequences containing the evolutionary/functional domains.

(2) *Classification step:* taking as a kernel the dot product between these vector representations to be used in conjunction with SVM.

In the proceeding sections, we describe both steps.

2.1 Feature extraction

Extracting structural domain information is shown to be tedious and remain unsolved completely. Therefore, we use a novel technique to extract the protein functional domains solely from amino acid sequence. We employ inter-domain linker region information. All amino acid appeared to be involved in the inter-domain linker region are identified and removed from the original membrane protein sequence. By conducting this step, the protein sequence will be shorter with only protein functional domains information, which may produce improved sequence matching scores. Domain linkers can play an essential role in maintaining cooperative inter-domain interaction [11].

The prediction of the inter-domain linker regions is made by using linker index deduced from a data set of domain/linker segments from SWISS-PROT database [12]. DomCut developed by Suyama et al. [13] is employed to predict linker regions among functional domains based on the difference in amino acid composition between domain and linker regions. Following [13], we defined the linker index S_i for amino acid residue i and it is calculated using the formula $S_i = -\ln\left(\frac{f_i^{Lin\ ker}}{f_i^{Domain}}\right)$ Where $f_i^{Lin\ ker}$ is the frequency of amino acid residue i in the linker region and f_i^{Domain} is the frequency of amino acid residue i in the domain region. A negative value of S_i indicates that the amino acid preferably exists in a linker region and therefore, we remove it from the protein sequence of interest. This step will result in significant downsizing of the protein sequence without compromising its generality.

2.2 Membrane protein sequence matching algorithm

Once the inter-domain linkers regions are removed and the short protein sequence contains potential functional domains is prepared, we measure the sensitivity of these short protein sequences against the membrane protein sequences of interest. The matching algorithm is illustrated as follows:

Let us assume that we would like to measure the matching between protein sequence 1 (s1: the original sequence which we would like to know its membrane type) and the protein sequence 2 (s2: the amino acid sequence consist of the predicted functional domains). To illustrate the algorithm, let us consider s1 and s2 as follow:

s_1	**Index**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	Amino Acid	S	V	Y	D	A	A	Q	L	T	A	D	V	K	K	D	L	R	D

s_2	**Index**	0	1	2	3	4	5	6	7	8	9	10	11
	Amino Acid	V	L	D	A	A	S	Q	N	K	K	A	S

To compare s1 to s2, let us assume that the first row presents the index of the amino acid and the second row presents the amino acid sequence. The algorithm starts by picking the first amino acid S (position 0) in s1 and search for a similar amino acid in s2. In this case, the amino acid S exists first at position 5 in s2. Starting at positions 0 (in s1) and 5 (in s2) and moving one amino acid at time to keep track if the following amino acids in the two sequences are matched i.e. the amino acid at index 1 (in s1) is compared to the amino acid at index 6 (in s2), the amino acid at index 2 (in s1) is

compared to the amino acid at index 7 (in s2), etc. The matching stops if 5 consecutive mismatches are found or when there is no more amino acids to compare to. The result of the matching is the following sequence S − − − −A− i.e s1[0] = s2[5] and s1[5] = s2[10]. The score of the match (S : A) is calculated using BLOSUM62 scoring matrix. A penalty of four mismatched is imposed in this case. At this point, the algorithm continues searching to find if there is another occurrence of S in s2. It finds another occurrence at position 11. It does the same processing mentioned above. Since there is no more amino acid remains in s2, the matching stops. The algorithm proceeds to the second amino acid V (in s1) and repeats the same procedures mentioned. It finds that V exists in position 0 in s2. Following the similar matching steps we obtained is V−DAA−−−−−. In this case, the algorithm stops when 5 mismatches are found and the corresponding score is calculated. The algorithm then proceeds to the third amino acid in s1 and does the same processing. When all the amino acids in s1 are processed, the matching between s1 and s2 is done and all the corresponding scores are accumulated to create the final matching score between the 2 proteins. The matching algorithm is illustrated in Fig. 2.

Algorithm 1: Protein Matching Algorithm

Data: - Files F_1 and F_2 containing the protein sequences
- 20×20 *scoring matrix* (BLOSUM62)
- Number of mismatches ($noMismatch$)

Result: matching scores between the protein sequences in F1 and F2

```
1  begin
2      Let P1 be an array of size n1 holding the proteins in F1
3      Let P2 be an array of size n2 holding the proteins in F2
4      for x = 0 ⟶ n1 do
5          for y = 0 ⟶ n2 do
6              s1 = P1[x]
7              s2 = P2[y]
8              pos2 = 0
9              for every amino acid c1 in s1 do
10                 - let pos1 be the index of c1 in s1;
11                 while true do
12                     - find the first occurrence of c1 in s2 after position pos2;
13                     - let pos2 be the index of c1 in s2;
14                     - if c1 is not found in s2 after position pos2, consider
                       the following amino acid from s1 → goto line 9;
15                     - loop across s1 and s2 starting at pos1 and pos2
                       respectively begin
16                         compare, one-by-one, amino acids from s1 and s2;
17                         if 2 consecutive amino acids in s1 match 2
                           consecutive amino acids in s2 then
18                             find their matching value v in the scoring
                               matrix;
19                             score += v;
20                         else
21                             find their matching value v in the scoring
                               matrix;
22                             score += v - penalty;
23                         stop when the number of consecutive mismatches =
                           noMismatch;
24                     end
25                     - increment pos2 by 1;
26                     - goto line 11;
27 end
```

Figure 2: Protein matching algorithm.

One significant characteristic of any algorithm is its computational efficiency. In this respect, the protein matching algorithm complexity is O(n1 × n2 × L1 × L2). Where n1 and n2 are the number of protein sequences in set F1 and set F2, respectively. L1, L2 are the length of the longest protein sequence in the set F1 and length of the longest protein sequence in the set F2, respectively.

It is worth noting that when comparing any two proteins, the matching algorithm stops executing when 5 consecutive mismatches occur (5 different consecutive amino acids). Given this technique and given the nature of proteins, it is very rare to visit all amino acids in two proteins. Therefore, the worst case complexity is rarely achieved. Moreover, the number of mismatches (currently 5) is a variable that the user could change it. If we set the value to 1, then the matching algorithm will attempt to find the exact matches between the two protein sequences. By varying the mismatch value, we can efficiently find the best matches between proteins.

Following the protein sequence matching step, all proteins will be represented by feature vectors which will allow us to take as a kernel the dot product between these vector representations to be used in conjunction with SVM.

2.3 Classification

The problem is basically formulated as multi-class classification problem: both training and testing sets contain membrane protein sequences belong to any of the eight types. This representation is combined with SVM to classify between the eight sets. The SVM algorithm addresses the general problem of learning to discriminate between positive and negative examples of a given class of n-dimensional vectors. In order to discriminate between the protein membrane types, the SVM learns a classification function from a set of positive examples $\chi+$ and set of negative examples $\chi-$. The classification function takes the form:

$$f(x) = \sum_{i:x_i \in \chi_+} \lambda_i K(x, x_i) - \sum_{i:x_i \in \chi_-} \lambda_i K(x, x_i) \qquad (1)$$

where the non-negative weights λ_i are computed during training by maximizing a quadratic objective function and the function $K(.,.)$ is called a kernel function [14]. Any new sequence x is then predicted to be positive if the function $f(x)$ is positive. More details about how the weights λ_i are computed and the theory of SVM can be found in [15]-[16].

3 Results and Discussion

The protein sequences used in this work was collected and published by Chou et al. [6]. The datasets are collected from the Swiss-Prot database at (version 51.0). To insure a high-quality of the benchmark dataset, the data were screened strictly according to the following criteria and order:

(1) Sequences annotated with ‘‘fragment’’ were excluded. Any sequence with less than 50 amino acid residues was excluded because it might just be fragments.

(2) Sequences annotated with ambiguous or uncertain terms, such as ‘‘potential’’, ‘‘probable’’, ‘‘probably’’, ‘‘maybe’’, or ‘‘by similarity’’, were removed for further consideration.

(3) For the sequences made it through the screening procedures in (1) and (2) which have clear experimental annotations, those annotated with ‘‘membrane protein’’ were stored in the membrane protein reservoir Datasetmem.

(4) Eight different membrane protein types were found in Datasetmem; to reduce the homology bias, a redundancy cutoff was operated to winnow those sequences which have ≥80% sequence identity to any other in a same membrane type.

Finally, we obtained a dataset S contains 7582 membrane protein sequences belong to any of the eight membrane protein types. According to their experimental annotations, the 7582 membrane proteins can be further classified into eight subsets. The numbers of proteins thus obtained for the eight membrane protein types in the training dataset and the testing dataset are given in Table 1.

Table 1 : Number of membrane proteins in each of the eight types for the training and the testing datasets.

Type	Number of sequences in the training dataset	Number of sequences in the testing dataset
Single-pass type I	610	444
Single-pass type II	312	78
Single-pass type III	24	6
Single-pass type IV	44	12
Multipass	1316	3265
Lipid-chain-anchor	151	38
GPI-anchor	182	46
Peripheral	610	444
Total	3249	4333

In the first step of the experimental work we started by extracting the functional domain using Domcut method. Amino acids with a negative value of S_i are removed from the training protein sequences dataset. This step resulted in significantly downsizing all the protein sequence in the training dataset without compromising their generality. These new short sequences are expected to contain only functional

domains information. The process here is slightly different from Domcut approach. In Domcut, a threshold value of -0.1 was considered as shown by the horizontal line in Figure 3. In this case linker regions (1, 2,.. , 10) less than a threshold value of -0.1 are identified from the protein sequence of interest. DomMat however, considers a threshold value of 0. Different threshold values are tested and a value of 0 resulted in higher accuracy rate since it identifies more suspected linker regions.

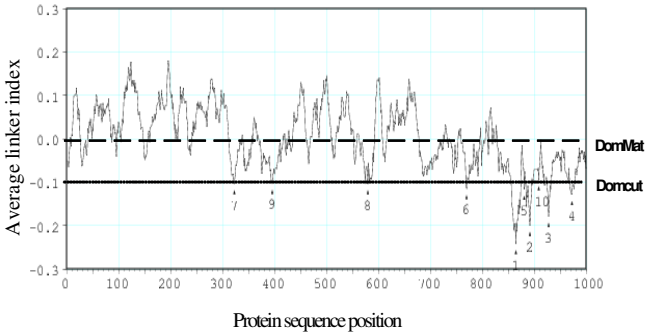

Figure 3: An example of linker preference profile generated using Domcut. In this case, linker regions (1, 2,.. , 10) less than a threshold value of -0.1 are identified from the protein sequence of interest. A horizontal line is also drawn at the averaged linker index value 0 which was used by DomMat.

In the second step, we match each protein sequence in the training and the testing set to the short protein sequences. We used a penalty of 5 and a mismatch size equal to 5. In the last step, we employ LIBSVM (Library for Support Vector Machines) to classify the testing set into eight protein membrane types. LIBSVM [17], is developed by Chang and Lin and it supports multi-class classification. In all experiments, Gaussian Radial Basis Function kernel (RBF kernel) was used, the RBF kernel allows pockets of data to be classified which is more powerful approach than simply using a linear dot product. The function has the form $k(x, x_i) = e^{-\gamma \|x - x_i\|^2}$, where $x, x_i \in X$ and $\gamma > 0$. In all of the experimental work, the scaling parameter γ was set to 0.001. The SVM algorithm is based on a sound mathematical framework and much of its power is derived from its criterion for selecting a separating hyperplane that maintains a maximum margin from any point in the training set [18].

The prediction results obtained are given in Figure 4. The corresponding accuracy results by MemType-2L [6] as one of the most accurate method so far are also shown. The overall accuracy success rate by MemType-2L was 91.6% while DomMat was able to achieve a comparable overall accuracy rate of 93.54%. The results clearly indicate that DomMat is indeed very useful in identifying membrane protein types.

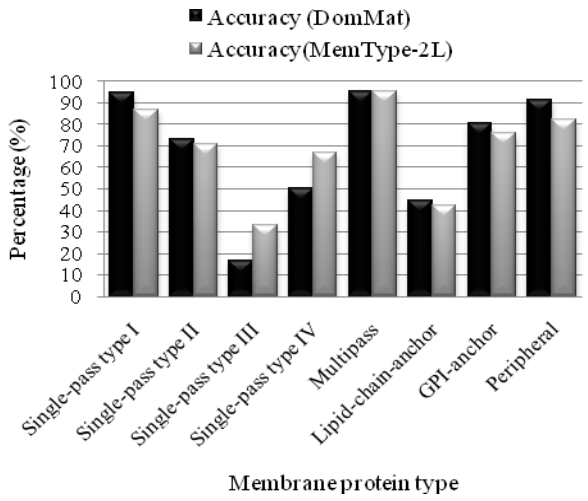

Figure 4: Success rate in identifying membrane protein types.

DomMat could also be used to distinguish between membrane and non-membrane protein. This experimental work is not included in this paper since membrane proteins consist of transmembrane proteins and anchored membrane proteins. The former contains one or more hydrophobic segments, and hence is relatively easily discriminated from nonmembrane proteins. We understand that the matching between two protein sequences may not necessarily lead to similar protein membrane type. However, it is evidence that we cannot ignore. Matching would allow one to infer homology and homologous sequences are usually structurally related. This relationship could lead to identify two membrane proteins from the same type. It's also worth to mention that, despite the novelty of functional domain extraction method, a more accurate technique to identify inter-domain linker regions is needed. Domcut has shown sensitivity (proportion of the total number of successfully predicted linkers against the total number of linkers) is 53.5% and selectivity (proportion of correct predictions in all predictions) is 50.1%.

4 Conclusions

DomMat is introduced in this paper. The method for predicting protein membrane types consists of two major steps: (1) Feature extraction step in which we represent each protein sequence by its sensitivity to a set of short amino acids sequences containing the evolutionary/functional domains, (2) Classification step which takes as a kernel the dot product between these vector representations to be used in conjunction with multi-class SVM. The results obtained were clearly indicated that DomMat is comparable to state-of-the–art methods and indeed very useful in identifying membrane protein types.

5 References

[1] T. Sandra, T. T. Hwee, C. M. Maxey, "Membrane proteins and membrane proteomics," *Proteomics*, 19: 3924 – 3932, 2008.

[2] W. Meng, Y. Jie, L. Guo-Ping, X. Zhi-Jie, C. Kuo-Chen, "Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition," *Protein Engineering, Design & Selection*, 6: 509–516, 2004.

[3] S. Hong-Bin, and C. Kuo-Chen, "Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types," Biochem. Biophys. Res. Commun., 334: 288-92, 2005.

[4] Y. D. Cai, K. C. Chou, "Predicting membrane protein type by functional domain composition and pseudo-amino acid composition," *J. Theor. Biol.*, 238:395-400, 2005.

[5] S. Hong-Bin, Y. Jie, and C. Kuo-Chen, "Fuzzy KNN for predicting membrane protein types from pseudo-amino acid composition," *J. Theor. Biol.,* 240: 9-13, 2006.

[6] C. Kuo-Chen, and S. Hong-Bin, "MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM," *Biochem. Biophys. Res. Commun.,* 360: 339-345, 2007.

[7] C. Chothia, A. Lesk, "The relation between the divergence of sequence and structure in proteins," *EMBO J*., 5:823-826, 1986.

[8] R. Doolittle, "Similar amino acid sequences: chance or common ancestry?, " *Science,* 214:149-159, 1981.

[9] M. Sierk, W. Pearson, "Sensitivity and selectivity in protein structure comparison," *Protein Sci.,* 13:773-785, 2004.

[10] T. Wood, W. Pearson, "Evolution of protein sequences and structures," *J. Mol. Biol.,* 291:977-995, 1999.

[11] R.S. Gokhale, and C. Khosla, "Role of Linkers in Communication between Protein Modules," *Curr. Opin. Chem. Biol.,* 4: 22-27, 2000.

[12] A. Bairoch, and R. Apweiler, "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000," *Nucleic Acids Res.,* 28, 45-48, 2000.

[13] M. Suyama, O. Ohara, "DomCut: prediction of inter-domain linker regions in amino acid sequences," *Bioinformatics,* 19: 673-674, 2003.

[14] H. Saigo, J. Vert, N. Ueda, T. Akutsu, "Protein homology detection using string alignment kernels," *Bioinformatics*, 20:1682-1689, 2004.

[15] V. N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[16] N. Cristianini, and J. Shawe-Taylor, *An introduction to Support Vector Machines*, Cambridge, UK: Cambridge University Press, 2000.

[17] C. Chih-Chung, L. and Chih-Jen. (2001) LIBSVM : a library for support vector machines. Available : http://www.csie.ntu.edu.tw/~cjlin/libsvm

[18] N. M. Zaki, S. Lazarova-Molnar, W. El-Hajj, P. Campbell, "Protein-protein interaction based on pairwise similarity," *BMC Bioinformatics*, 10:150, 2009.