Approximate Symbolic Pattern Matching for Protein Sequence Data

Bill C. H. Chang and Saman K. Halgamuge

Mechatronics Research Group, Department of Mechanical and Manufacturing Engineering, University of Melbourne, Melbourne, Victoria 3010, Australia

Abstract

In protein sequences, often two sequences that share similar substrings have similar functional properties. Learning of the characteristics and properties of an unknown protein is much easier if its likely functional properties can be predicted by finding the substrings already known from other protein sequences. The sequence pattern search algorithm proposed in this paper searches for similar matches between a pattern and a sequence by using fuzzy logic and calculates the degree of similarity from a sequence inference step. Proteins from 11 domain families are used for simulation and the result shows that the proposed algorithm is capable of identifying sequences that have a similar pattern compared to their family protein motifs.

Keywords: Sequence pattern matching, approximate sequence matching, protein sequences, fuzzy, data mining

1. Introduction

Mining of sequence data has many real world applications [1][2]. Transaction history of a bank customer, product order history of a company, performance of the stock market [3] and biological DNA data [4] are all sequence data where sequence data mining techniques are applied. In contrast to ordinary data set, sequence data are dynamic and order dependent. An example of a sequential pattern is "A customer who bought a Pentium PC nine months ago is likely to order a new CPU chip within one month" [1].

For symbolic sequential data, pattern matching can be considered as either (1) exact matching or (2) approximate matching [2]. Approximate matching is the finding of the most similar match of a particular pattern within a sequence. Quite often in real world data mining applications, exact patterns do not exist due to the large number of possible sequence combinations, and therefore, an approximate matching algorithm is required. Especially in the field of molecular biology, sequence patterns are described in an approximate way.

The analysis of protein and DNA sequence data has been one of the most active research areas in the field of computational molecular biology [5]. A DNA sequence contains genetic information, which include genes, regulatory regions, and many other unknown regions. Four different "bases" – A, T, C and G are the "building blocks" in DNA sequences and a DNA sequence is composed of a combination of these four bases in a linear form [6]. Computational molecular biology research in DNA sequences mainly concentrate on gene identifications, regulatory region identifications and specie to specie comparisons.

When a gene in a DNA sequence is "activated", a process called transcription starts and mRNA (a genetic material) is produced. The sequence of mRNA is the complement (i.e. A becomes T, C becomes G, G becomes C and T becomes A) of the gene it was transcribed from. mRNA is then translated (every 3 DNA bases is translated into 1 Amino Acid, which is the building block of proteins) into protein sequence which is then folded into a functional three dimensional structure [6].

The discovery of patterns within biological sequences can lead to significant biological discoveries. Sequence motif discovery algorithms can be generally categorized into 3 types: (1) String Alignment algorithms, (2) Exhaustive enumeration algorithms, and (3) Heuristic methods. Motif discovery is outside the scope of this paper and the motif patterns used for simulation are obtained from PROSITE database [7]. Please refer to [8] for a more detailed discussion on motif discovery algorithms.

It is important to have a pattern matching technique to identify the biological-significant patterns within an unknown biological sequence. This presence of significant patterns within an unknown sequence gives some indication of the likelihood of its functional properties. This paper discusses the possibility of implementing an approximate pattern matching algorithm based on a *fuzzy inference* technique. Background on sequence searching in protein database is briefly presented in Section 2. Section 3 summarises the method of Fuzzy Sequence Pattern Searching Algorithm between two sequences. Simulation results on molecular biology data and its discussion are presented in Section 4 and Section 5. Finally, Section 6 discusses some of the possible future developments.

2. Sequence Searching in Protein Database

Biologists often perform protein sequence searches due to the fact that similar sequences usually have similar functional properties [2]. When an unknown protein is sequenced, the scientist usually tries to get the "feel" for its functional properties by doing database searching. The functional properties of a protein can be scientifically determined using biological tests, however, the testing time period would be quite long if the scientist does not have any idea about the particular protein sequence. By doing a sequence search through protein sequence database (such as PROSITE [7], BLOCKS [9], PRINTS [10] and PFAM [11]), and if some similar sequences exist, the scientist would often test for the possibility of similar functional properties for the unknown sequences and hence, fast track the research.

Generally, there are two approaches of sequence searching in proteins: sequence alignment and sequence motif searching. Sequence alignment methods have two variations: local alignment and global alignment methods [2]. Local alignment method aims to align two sequences so that the similarity between the regions of the two sequences is maximised. Local alignment such as BLAST [12] is useful to determine the functional similarity between two sequences. On the other hand, global alignment methods aim to align two sequences so that the similarity between the two sequences is maximised (globally). Global alignment method is useful to determine the relevance between two sequences in terms of their inheritance. Alignment algorithm is usually implemented based on dynamic programming technique and variation of alignment methods can be achieved by manipulating the scores for matches, mismatches, gaps, etc [2].

Sequence motif searching techniques identify the existence of motifs within an unknown protein sequence. A Protein sequence motif, signature or consensus pattern, is a short sequence that is found within sequences of a same protein family [13]. PROSITE [7] is one of the protein motif database where scientist can search for occurrences of protein motifs in his/her unknown protein sequences.

Apart from searching for similar sequences, protein sequence searching can also potentially obtain the structural information about an unknown protein. A protein conformation is often described in terms of three structural levels: (1) the Primary Structure, which is a linear sequence of polypeptide chain (series of linked amino acids), (2) the Secondary Structure, which describes the path that the polypeptide backbone of the protein follows in space, and (3) the Tertiary Structure, which describes the organisation in three dimensions of all the atoms in the polypeptide chain [6]. A protein's sequence which is also known as the primary structure of the protein can often be used to predict its secondary structure [14]. However, the prediction of a protein's tertiary structure is still difficult at this stage.

In real world biological applications, most relevant sequences are "similar" instead of exactly the same. It is therefore useful to search sequences using fuzzy logic where approximate pattern searching can be implemented. Current approximate searching algorithms (such as Prosite Scan [15]) is rigid in a way that its definition of "similarity" is fixed. In the proposed algorithm, the user is able to define the meaning of "similarity" by adjusting the membership functions for sequence search. This way, a scientist is able to identify an unknown sequence's functional properties using the past experience and expertise.

3. Approximate Sequence Pattern Searching Algorithm

In *exact* sequence pattern matching problems, we aim to find a substring in text *T* that is exactly the same as the searching pattern *P*. In biological sequence data applications, exact patterns are rare, but sequences belonging to the same functional family usually have "similar" substrings within each of the sequences. Hence, there is a requirement of an *approximate* sequence pattern searching algorithm for biological data analysis.

The proposed algorithm aims to find a substring, *P'*, within a text, *T*, that is "most similar" to a searching pattern, *P*. A sequence data can be interpreted as a series of *events*, E_I, separated by their *event intervals*, I_{ij}. A sequence can be described as:

$$E_1 - I_{1,2} - E_2 - I_{2,3} - E_3 \ldots - I_{(n-1),n} - E_n$$

For example, the sequence *ATG* has three events *A*, *T* and *G*. The event intervals between *A* and *T*, and *T* and *G* are both zero (i.e. $I_{1,2} = I_{2,3} = 0$). The concept of *event intervals* is important when the searching pattern, *P*, contains *wild cards*. A wild card, usually represented by letter "*X*" in molecular biology literature, can match to any other symbols. For instance, sequences *AXC* and *ABC* are considered to be an "exact match" as *X* can be matched to *B* (in this case) or any of the possible symbols/events. Since *X* is a wild card and not an identified event for the search algorithm, the sequence *AXC* has only two events, *A* and *C* separated by an event interval of one.

A classifying type fuzzy system without defuzzification [16] is used for the proposed algorithm. There are four main steps in the proposed algorithm. Firstly, a searching pattern (string), *P* is decomposed to obtain *events* and *event intervals*. Then, the obtained events and event intervals are fuzzified in the Sequence Fuzzification step. A Sequence Inference step follows to determine the sequences that are "similar" to the searching pattern *P*. Finally, a sequence search is conducted to determine the "similarity" between a text *T* and the pattern *P*. An overview of the algorithm procedure is shown in Figure 1.

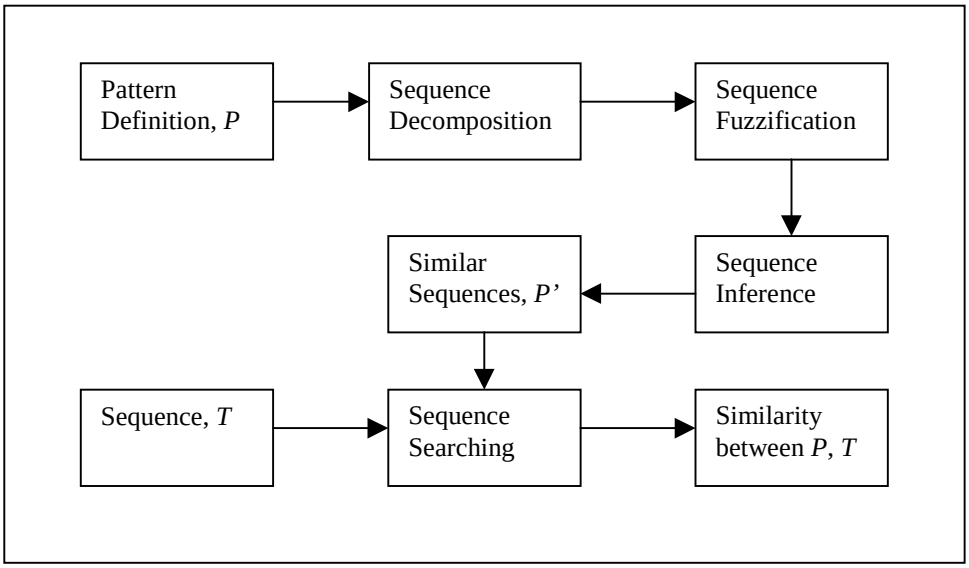

Figure 1. Overview of Approximate Sequence Searching Algorithm.

3.1 Sequence Decomposition

In this step, the searching pattern *P* is decomposed to obtain events and event intervals. Events in the sequence are identified and stored in an *event distribution matrix*. From this event distribution matrix, event intervals for all events can be calculated.

An event can consist of one or more symbols/characters, and *event width* is the number of symbol(s)/character(s) of an event. "*C*" is an event with event width equals to one, whereas "*CT*" is an event with event width of two. The number of decomposition *level* needed corresponds to the *event width* of an event. First level decomposition identifies each character as an event and *k-th level* decomposition identifies events with an event width of *k*.

An example of event identification for sequence *CTGACAG* and its event distribution matrix is shown in Figure 2.

3.2 Sequence Fuzzification

The searching pattern *P* is fuzzified by applying fuzzification techniques to the events and event intervals obtained from the previous step. In the fuzzification step, fuzzy membership functions of events and event intervals are generated. There are three fuzzy variables: event content, event interval, and total number of events.

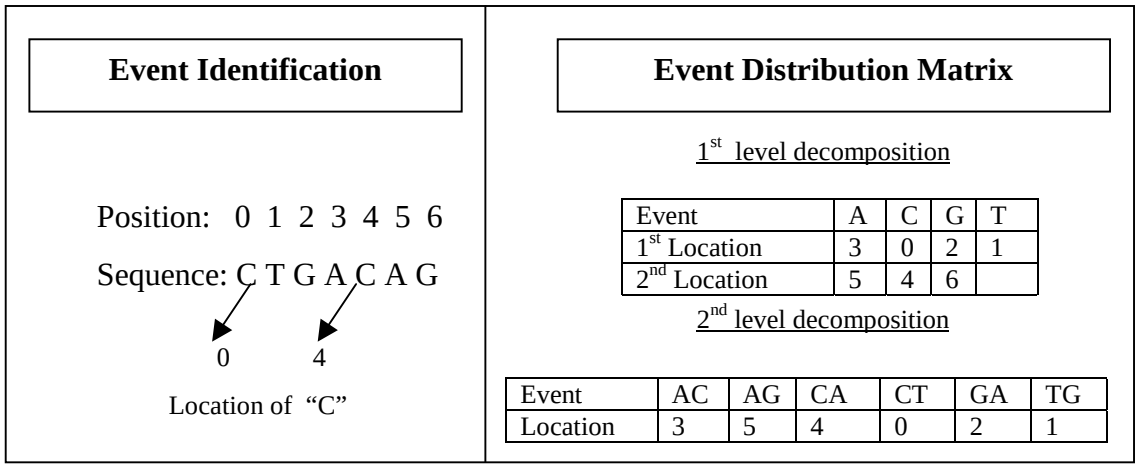

Figure 2. Event identification and event distribution matrix for sequence *CTGACAG*.

3.2.1 Event Content Fuzzy Membership Functions

An event can be fuzzified based on its content, or character(s)/symbols(s) presented. The assignment of the fuzzy membership functions is dependant on the requirement of the specific task or expert knowledge. For example, if event *TG* in pattern *P*, is considered to be important even if only one of the symbols exists. We can assign *XG* and *TX* a value of 0.5, where symbol *X* can match to any other symbols. A fuzzy membership function for this event can be generated as shown in Figure 3.

3.2.2 Event Interval Fuzzy Membership Functions

The length of an event interval can be fuzzified to represent *Long*, *Medium*, *Short*, or any other linguistic terms. This concept is useful in biological applications since number of wild cards varies for many protein family motifs. An example of fuzzy membership functions for event interval is shown in Figure 4.

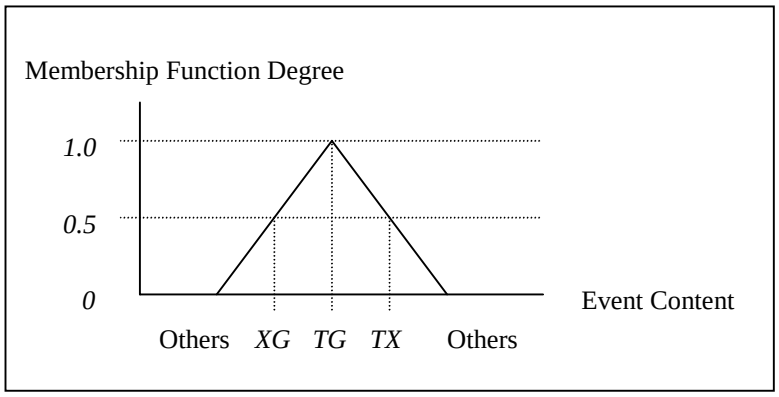

Figure 3. Fuzzy membership function for event *TG*.

3.2.3 Total Event Fuzzy Membership Functions

In some applications or tasks, we would like to detect a sequence data even if some of the events in the searching pattern are non-existent in the sequence data. Especially in the biological data where two proteins share "enough similarity" in sequences may have similar functional properties. This is also known as the "First fact of biological sequence analysis" [2]: "In bio-molecular sequences (DNA, RNA, or amino acid sequences), high sequence similarity usually implies significant functional or structural similarity [2]. For example, the sequence *ATGCA* and *ATGCC* may have the same functional property even though *ATGCC* only has four events of *ATGCA*. An example of total event fuzzy membership function for *P* = *ATGCA* is shown in Figure 5. Here the variable *Tot*, is used to describe the total number of events.

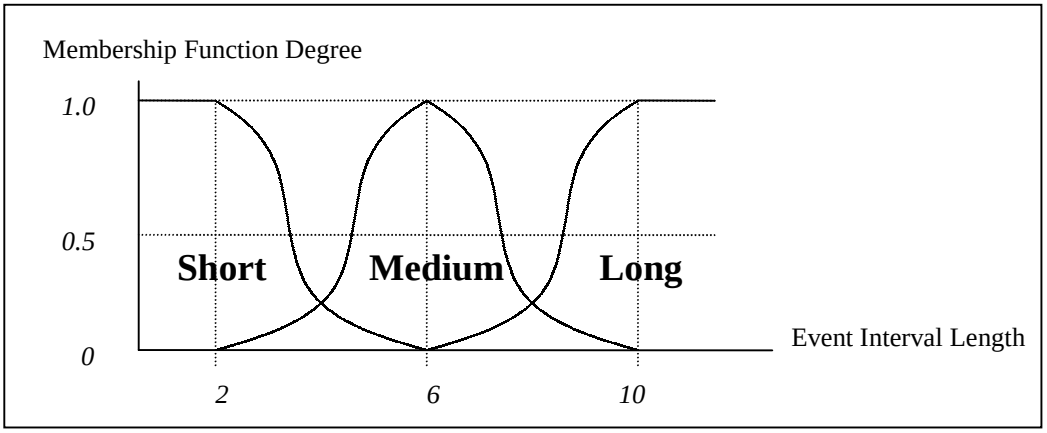

Figure 4. Fuzzy membership function for event interval: *Long*, *Medium*, and *Short*.

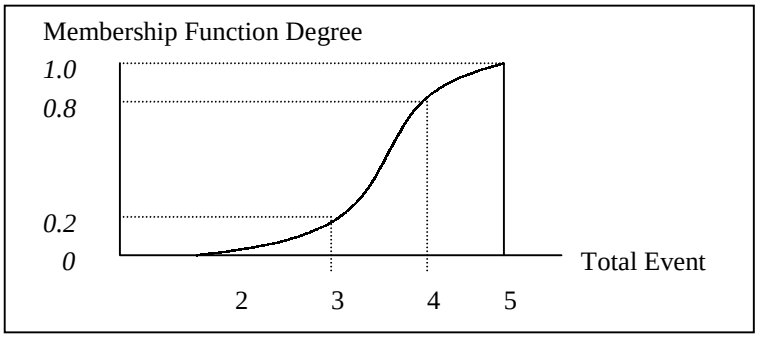

Figure 5. Total event fuzzy membership function for *P = ATGCA*.

3.3 Sequence Inference

This step generates an array of sequences *P'* that are "similar" to the searching pattern *P*. The degree of similarity is determined by the fuzzy rule:

R_i: IF event E_1 occurs AND event E_2 occurs AND event interval between E_1 and E_2 is I_1 AND … event E_{n-1} occurs AND event E_n occurs AND event interval between E_{n-1} and E_n is I_{n-1}, AND the total number of events is *Tot*, THEN Pattern P'_i is similar to *P* with a degree Y_i.

Where Y_i = T-norm($\mu(E_1)$, $\mu(E_2)$, …, $\mu(E_n)$, $\mu(I_1)$, $\mu(I_2)$, …, $\mu(I_{n-1})$, $\mu(Tot)$)

The T-norm used can be multiplication and the choice of functions will depend on the need of specific applications.

3.4 Sequence Searching

The array of similar sequences *P'* obtained from the previous step is then used for the determination of similarity between a sequence text *T* and a searching pattern *P*. Each P'_i is compared with sequence *T* as an exact matching problem and if P'_i exists in *T*, then the similarity between *P* and *T* is Y_i. Since the sequence *T* can match to many of the sequences in *P'*, the similarity between *T* and the searching pattern *P* is determined as:

$Y = F(Y_j)$, for P'_j exists in *T*.

and the function F is Maximum.

4. Simulation Results

In this simulation, firstly, artificially created sequence data is used to demonstrate the use of the proposed algorithm. Sequence data of eleven protein families from Swiss-Prot Protein Database [15] are then used to demonstrate the applicability of the proposed algorithm on real biological applications.

4.1 Simulation with Artificial Data

Artificially generated sequences shown in Figure 6 are used to demonstrate the use and scope of the proposed algorithm.

In this simulation, we will demonstrate the use of the proposed algorithm by using the event interval membership functions, event content membership functions, and the total event membership functions in

our sequence searching pattern *P*. We start our simulation from searching sequences that contain the pattern *P*, where:

P = *A*, then some small number of wild cards, and then another *A*, (i.e. *A*-x(0,6)-*A*)

The event interval between the two *A*s is *short* with a membership function as shown in Figure 4. This search found 14 valid sequences with a membership function value greater than zero. The result of search is presented as " membership function value/sequence number": [0.8/2; 0.1/3; 1/4; 1/7; 0.8/8; 1/9; 1/10; 0.8/11; 0.2/13; 0.8/14; 1/15; 0.8/17; 1/18; 1/20].

1	ABCDEFGHIJFDAMNVBZGFGA
2	FASDFASDFSAFHXVCNB
3	NBVCXBGADTGBHADFDDYUTUKD
4	HGAGFDVRYTHAGFTADFADBNBZVC
5	OFLEPFKSSSFFASDF
6	ZMVKFIEJKFAVCGG
7	KFDAFEFADFAGAFGADATD
8	KKFASDFADSFTRTQR
9	KDKKQDAFADAGAFDGA
10	KFAFKDAOEOAFAAGFGAGADYRR
11	FAUQFASDFKNCDDSAEOFDFS
12	DAIIHRIEPDNFDKPSEJDIFJDLKE
13	DFAFDFHAGKDKFKPPPPDFAJNFE
14	FASDFUWADFIANCZSAFOIE
15	PAADSIOPOQERIAFJDNAFJAFAFU
16	HSFDDFAUERUIQJFJASDFNDFA
17	RTYSDFHASDORQERIHAFDFAS
18	AFADSFHAFASDFAVDSFA
19	DFASDFEURERUEHS
20	KASFNVKDADFASDFOE

Figure 6. 20 artificially generated sequences.

Let us define an arbitrary symbol "*$*" which has a membership function shown in Figure 7. The searching pattern *P* is extended to include two more "*$*" symbols after the second *A*. So the searching pattern becomes:

P = *A*, then some small number of wild cards, and then another *A*, then two "*$*" symbols

Here, we try to demonstrate the use of event membership functions. This search yielded 11 valid sequences: [0.2/2; 0.05/3; 0.4/4; 0.4/7; 0.2/8; 0.5/9; 0.2/11; 0.05/14; 0.25/15; 0.25/18; 0.25/20]. Then, we add another symbol *F* at the end of the searching pattern *P*:

P = *A*, then some small number of wild cards, and then another *A*, then two "*$*" symbols, then *F*.

With the new search pattern *P*, there are 5 valid sequences: [0.2/2; 0.2/8; 0.2/11; 0.25/18; 0.25/20]. This pattern has five events: *A*, *A*, *$*, *$*, and *F*. If we want to detect sequences that have the pattern *P* with at least four of the five events present, the total number of valid sequences becomes 16. These are: [0.2/2; 0.025/3; 0.2/4; 0.25/5; 0.2/7; 0.2/8; 0.25/9; 0.2/11; 0.25/13; 0.025/14; 0.125/15; 0.125/16; 0.25/17; 0.25/18; 0.125/19; 0.25/20].

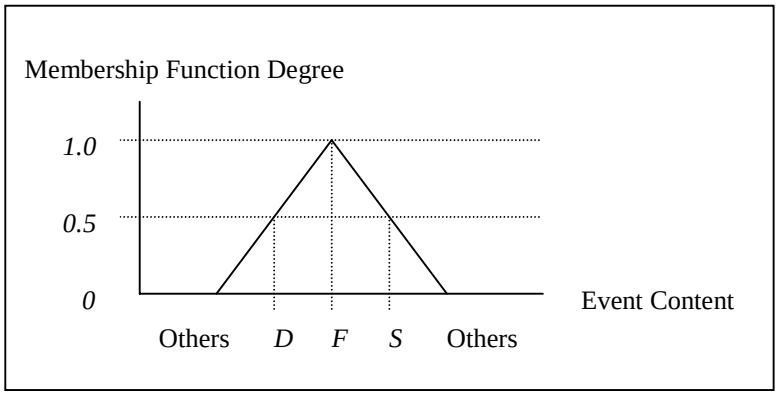

Figure 7. Fuzzy membership function for event "*$*".

4.2 Simulation with Protein Sequence Data

C2H2 Zinc Finger proteins are used for demonstration of the proposed algorithm and this simulation is done using PROSITE database release 39 [17]. "C2H2 are nucleic acid-binding protein structures first identified in the Xenopus transcription factor TFIIIA. These domains have since been found in numerous nucleic acid-binding proteins. A zinc finger domain is composed of 25 to 30 amino-acid residues. There are two cysteine or histidine residues at both extremities of the domain, which are involved in the tetrahedral coordination of a zinc atom"[17]. In PROSITE, the motif pattern for C2H2 Zinc Finger proteins is:

C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H

The symbol x(i,j) represents the existence of i to j number of wild cards, whereas x(i) means that there are i number of wild cards. The section [LIVMFYWC] represents one of the "*LIVMFYWC*" symbol is represented. The process of fuzzy pattern searching starts with the Sequence Decomposition step described in Section 3.1.

4.2.1 Sequence Decomposition

Since our searching pattern P = C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H, does not contain substrings of multiple characters, 1st level decomposition is used. Five events and four event intervals are identified:

Events: $E_1 = C$; $E_2 = C$; E_3 = [LIVMFYWC]; $E_4 = H$; $E_5 = H$.
Event Intervals: $I_{1,2}$ = 2,3,4; $I_{2,3}$ = 3; $I_{3,4}$ = 8; $I_{4,5}$ = 3,4,5;

4.2.2 Sequence Fuzzification

The events and event intervals are fuzzified according to the method described in Section 3.2. The membership functions implemented are shown in Figure 8. The event contents for single character symbol (*C*, and *H*) are not fuzzified in this simulation. For event E_3, since no preference of symbols is given, all symbols inside the bracket is given a membership degree of 1. These membership functions can be modified according to the specific needs of the user.

4.2.3 Sequence Inference

The inference rule used for this pattern is:

> IF event E_1, E_2, E_3, E_4 and E_5 occur, AND their event intervals are $I_{1,2}$, $I_{2,3}$, $I_{3,4}$ and $I_{4,5}$, AND the total number of events is *Tot*, THEN Pattern P'_i is similar to P with a degree Y_i.

Where Y_i = T-norm($\mu(E_1)$, $\mu(E_2)$, $\mu(E_3)$, $\mu(E_4)$, $\mu(E_5)$, $\mu(I_{1,2})$, $\mu(I_{2,3})$, $\mu(I_{3,4})$, $\mu(I_{4,5})$, $\mu(Tot)$)

The T-norm used is multiplication.

4.2.4 Sequence Searching

Since a protein sequence T can match to many of the sequences in P', the similarity between T and the searching pattern P is determined as:

> $Y = F(Y_j)$, for P'_j exists in T.

and the function F is Maximum.

The simulation results show that the proposed algorithm identified 416 out of 418 Zinc Finger Protein Sequences. The two protein sequences already experimentally identified as Zinc Finger Proteins but not detected by the proposed algorithm are:

- YMDFVAAQCLVSISNRAAPEHGVAPDAERLRLPEREVTKEHGDPGDTWKDYCTLVTIAKSL LDLNKYRPIQTPSVCSDSLESPDEDMGSDSDVTTESGSSPSHSPEERQDPGXAPSPLSLLHPGV AAKGKHASEKRHK

- HIAHHTLPCK**C**PI**C**GKP**F**APWLLQG**H**IRT**H**TGESPSVCQHCNRAFA

The first sequence does not seem to have a subsequence that is similar to this common zinc finger motif, whereas the second sequence can be easily identified by modifying the membership function for $I_{3,4}$. Simulation result for other 10 randomly selected protein families (from PROSITE) is shown in Table 1.

5. Discussion

The simulation shows that the proposed algorithm is capable of doing approximate sequence pattern searching with a high success rate. From the simulation of C2H2 Zinc Finger Proteins, we found that 1 out of the 418 sequences does not have similarity with the motif C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H. This shows the variety of biological sequences. Although they are all catagorised as C2H2 Zinc Finger proteins, they may still have different sequence structures [18]. One way to overcome this problem may be to adopt a more general motif sequence. However, this strategy may result in a higher percentage of false positives.

Simulation results from Table 1 shows that the proposed algorithm can detect motif patterns as good as the PrositeScan program or better (PS00028, PS00150, PS00605). Detection of all protein sequences in a protein family is sometimes not possible due to the reason that some protein sequences only have part of the motif.

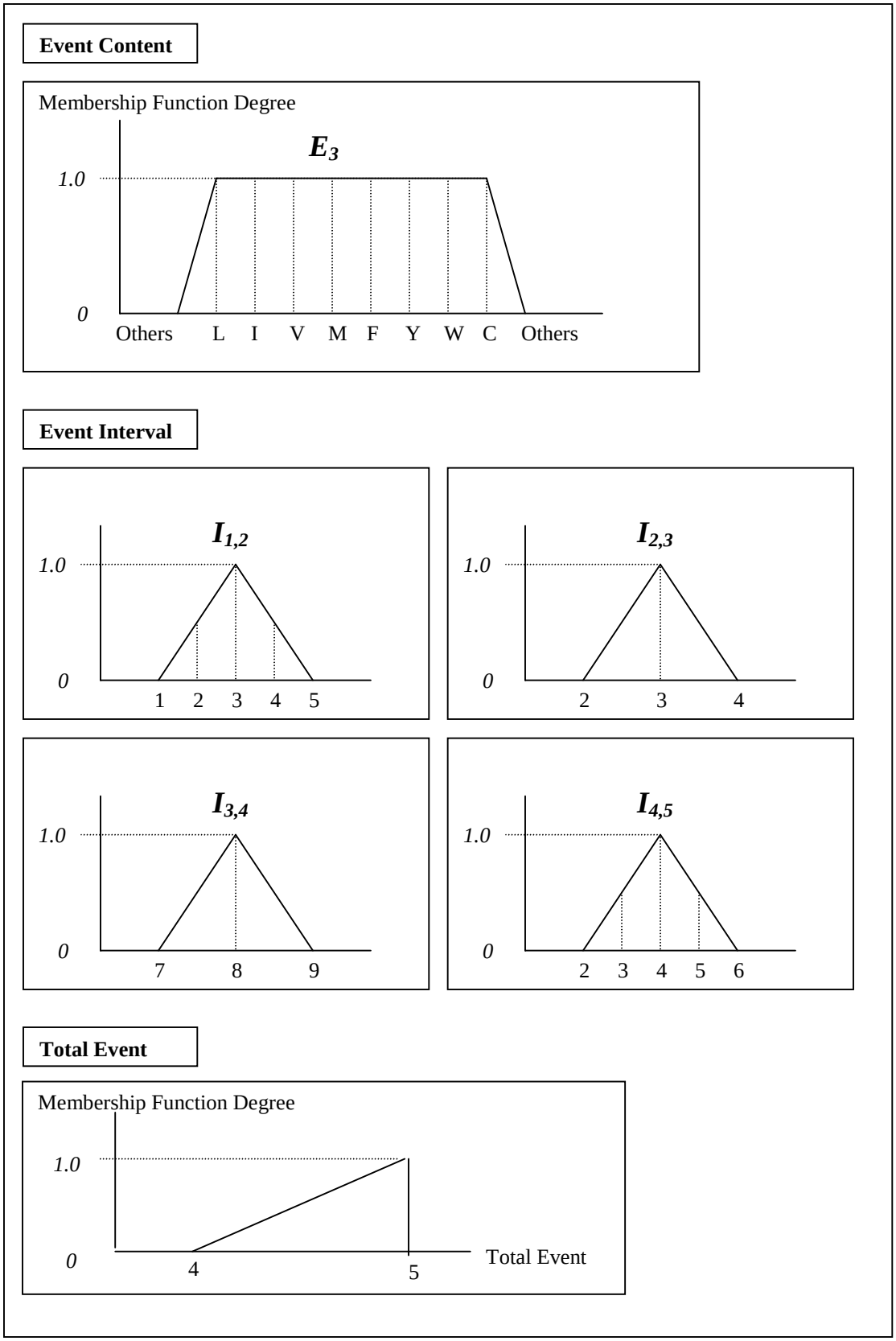

Figure 8. Fuzzy membership functions for Zinc Finger Protein Motif.

The fuzzy membership functions can be customised to suit a specific need. In the simulation, they are designed so that "similar" sequences will have membership function degrees of greater than zero. Of course, the term "similar" is fuzzy, and its definition can be different from one user to another. For example, in a sequence motif detection application for an unknown sequence where no motifs are found, the membership function can be designed so that all "similar" patterns have a membership function degree close to one. This way, a remotely similar pattern would have a membership degree greater than zero and this pattern can be detected.

The proposed algorithm has been successfully applied to number of protein sequence motifs searching with motifs presented in the form of "Regular Expression". This algorithm can also be extended to tackle the searching problems where sequence pattern is defined in an approximate way, as for example, the pattern for promoter sequences in Escherichia coli (E. coli) has been described as [19]:

1. all known E. coli promoters that use $E\sigma^{70}$ have at least two of the three most conserved bases in the –10 region. (TataaT, where capital letters represent "most conserved" bases)
2. all promoters have at least one of the most highly conserved TTG residues in the –35 region
3. those promoters with poor homology to the consensus in the –35 regions are frequently positively controlled by dissociable activators, and
4. the promoters used by E. coli $E\sigma^{32}$ during the heat shock response have similar –35 region sequences, but very different –10 region sequences.

These four rules provide the basic searching pattern for promoter sequences and the identification of promoters in a DNA sequence is still one of the most difficult tasks in molecular biology research due to its "approximate" nature.

6. Conclusion

This paper presents an approximate sequence pattern searching algorithm and it was successfully implemented to perform a protein sequence search for 11 randomly selected protein families. Simulation results show that the proposed algorithm is useful in identifying patterns with variable length wild cards and sequence symbol substitutions. The number of symbols presented in a pattern can also be fuzzified to adjust for the variations in the real world applications.

The author is currently working on an adaptive algorithm which "tunes" the membership functions to improve the classification performance. An extension of the proposed algorithm is expected to search for multiple patterns generated from the protein motif extraction algorithm proposed in [8]. The author is also looking to apply this algorithm to the problems of promoter sequence identification.

7. References

[1] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Academic Press, 2001.

[2] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[3] E. Gately. *Neural Network for Financial Forecasting*. John Wiley & Sons Inc, 1996.

[4] S. Misener and S. A. Krawetz. *Bioinformatics: Methods and Protocols*. Human Press Inc, 2000

[5] S. Salzberg, D. Searls and S. Kasif. *Computational Methods in Molecular Biology*. Elsevier, 1998.

[6] B. Lewin. *Genes V*. Oxford University Press, 1994.

[7] K. Hoffman, P. Bucher, L. Falquet and A. Bairoch. The PROSITE database, its status in 1999. *Nucleic Acids Research* 27:215-219, 1999.

[8] Bill C. H. Chang and Saman K. Halgamuge. Protein Motif Extraction with Neuro-Fuzzy Optimisation. *Bioinformatics*, issue 7, vol 18, 2002.

[9] S. Henikoff, J. Henikoff and S. Pietrokovski. Blocks: a non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, 15: 471-479, 1999

[10] T. Attwood, D. Flower, A. Lewis, J. Mabey, S. Morgan, P. Scordis, J. Selley, and W. Wright. PRINTS prepares for the new millenium. *Nucleic Acids Research*, 27: 220-225, 1999.

[11] A. Bateman, E. Birney, R. Durbin, S. Eddy, R. Finn and E. Sonnhammer. Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acids Research* 27:260-262, 1999.

[12] S. Altschul, etal. "Basic Local Alignment Search Tool", in *Journal of Molecular Biology*. Vol. 215, 1990. p.403-410.

[13] P. Bork and E. Koonin. Protein sequence motifs. *Curr. Opin. Struct. Biol.* 6:366-376, 1996.

[14] B. Berger and M. Singh. An Iterative Method for Improved Protein Structural Motif Recognition. In *Proceedings of RECOMB 1997*, p. 37-46, New Mexico, USA, 1997.

[15] Prosite Scan Internet site. http://npsa-pbil.ibcp.fr/cgi-bin/npsa_automat.pl?page=npsa_prosite.html

[16] S. K. Halgamuge. "Self Evolving Neural Network for Rule Based Data Processing", in *IEEE Transactions on Signal Processing*, November, 1997.

[17] Swiss-Prot Protein Database. http://au.expasy.org/sprot/

[18] S. Bohm, D. Frishman and H. Mewes. "Variations of the C2H2 zinc finger motif in the yeast genome and classification of yeast zinc finger proteins", *Nucleic Acid Research*, Vol. 25, no. 12, June 1997. p. 2464-2469.

[19] W. McClure. "Mechanism and Control of Transcription Initiation in Prokaryotes", *Annual Review in Biochemistry*, Vol. 54, 1985. p. 171-204.

PROSITE ID	Description	Pattern	Number of Protein Sequences in this family	Number of Protein Sequences detected by PrositeScan [13]	Number of Protein Sequences detected by Fuzzy Pattern Searching Algorithm
PS00028	Zinc Finger	C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H	418	412	416
PS00100	Chloramphenicol acetyltransferase active site	Q-[LIV]-H-H-[SA]-x(2)-D-G-[FY]-H	20	20	20
PS00110	Pyruvate kinase active site signature	[LIVAC]-x-[LIVM][LIVM]-[SAPCV]-K-[LIV]-E-[NKRST]-x-[DEQHS]-[GSTA]-[LIVM]	67	56	56
PS00120	Lipases, serine active site	[LIV]-x-[LIVFY]-[LIVMST]-G-[HYWV]-S-x-G-[GSTAC]	72	63	63
PS00150	Acylphosphatase signature 1	[LIV]-x-G-x-V-Q-G-V-x-[FM]-R	29	26	29
PS00230	Neuraxin and MAP1B proteins repeated region signature	[STAGDN]-Y-x-Y-E-x(2)-[DE]-[KR]-[STAGCI]	3	3	3
PS00250	TGF-beta family signature	[LIVM]-x(2)-P-x(2)-[FY]-x(4)-C-x-G-x-C	127	117	117
PS00272	Snake toxins signature	G-C-x(1,3)-C-P-x(8,10)-C-C-x(2)-[PDEN]	188	175	175
PS00300	SRP54-type proteins GTP-binding domain signature	P-[LIVM]-x-[FYL]-[LIVMAT]-[GS]-x-[GS]-[EQ]-x(4)-[LIVMF]	57	48	48
PS00411	Kinesin motor domain signature	[GSA]-[KRHPSTQVM]-[LIVMF]-x-[LIVMF]-[IVC]-D-L-[AH]-G-[SAN]-E	62	62	62
PS00605	ATP synthase c subunit signature	[GSTA]-R-[NQ]-P-x(10)-[LIVMFYW](2)-x(3)-[LIVMFYW]-x-[DE]	81	76	77

Table 1. Simulation result for 11 randomly selected protein families.