FlexSLiM: a Novel Approach for Short Linear Motif Discovery in Protein Sequences

Xiaoman Li*
Burnett School of Biomedical Science
University of Central Florida
Orlando, FL, 32816, USA
1-407-823-4811
xiaoman@mail.ucf.edu

Ping Ge*
Department of Computer Science
University of Central Florida
Orlando, FL, 32816, USA
1-407-823-4811
pge@knight.ucf.edu

Haiyan Hu#
Department of Computer Science
University of Central Florida
Orlando, FL, 32816, USA
1-407-823-4811
haihu@cs.ucf.edu

ABSTRACT
Short linear motifs are 3 to 11 amino acid long peptide patterns that play important regulatory roles in modulating protein activities. Although they are abundant in proteins, it is often difficult to discover them by experiments, because of the low affinity binding and transient interaction of short linear motifs with their partners. Moreover, available computational methods cannot effectively predict short linear motifs, due to their short and degenerate nature. Here we developed a novel approach, FlexSLiM, for reliable discovery of short linear motifs in protein sequences. By testing on simulated data and benchmark experimental data, we demonstrated that FlexSLiM more effectively identifies short linear motifs than existing methods. We provide a general tool that will advance the understanding of short linear motifs, which will facilitate the research on protein targeting signals, protein post-translational modifications, and many others.

CCS Concepts
• Information systems → Desktop search • Theory of computation → Pattern matching

Keywords
Short linear motif; protein sequences; frequent pattern mining; deterministic finite automaton.

1. INTRODUCTION
Short linear motifs (SLiMs) are peptide patterns that are typically 3 to 11 amino acid long. They are highly abundant in protein sequences [1-3]. For instance, it is estimated that hundreds of SLiMs and well over a million SLiM instances exist in human proteins [2, 3]. These SLiMs mediate various protein activities such as activation or deactivation of proteins, modifying proteins through post-translational modifications, serving as target sites for cleavage by proteases, and targeting proteins to specific subcellular locations, etc. [1]. Systematic identification of SLiMs is thus essential to understand protein regulation and protein behaviours. Despite their abundance and versatility, SLiMs are largely unexplored. For example, only 267 SLiMs are annotated in the Eukaryotic Linear Motif (ELM) database [1], the largest database that store experimentally validated SLiMs.

Experimental and computational methods have been developed for the discovery of SLiMs. Experimentally, the determination of SLiMs and their instances has been greatly aided by the use of antibodies [4, 5]. It is through these experiments that we start to know certain nature of SLiMs. However, experimental approaches are time-consuming and expensive. In addition, the interaction of SLiM instances with their partners are often subtle and transient, and thus are difficult to detect by experiments. Therefore, computational methods are indispensable for the identification of SLiMs. Based on SLiMs identified in experimental studies, several computational methods are developed to predict new instances of known SLiMs [1, 6-11]. These methods in general represent SLiMs in different ways (such as regular expression or position weight matrix) [9, 12, 13] and use different filters to remove segments that unlikely contain SLiM instances (such as globular domains) [14]. They have been shown successful in identifying meaning instances of known SLiMs [9, 12]. However, these methods are circumscribed by the limited number of known SLiMs [1, 8]. To de novo identify SLiMs, a few computational methods are developed [10, 11, 15-22]. Many of these methods [15, 16, 19, 20] construct elementary patterns from input protein sequences and then extend elementary patterns into SLiMs by concatenating pairs of similar elementary patterns. Here elementary patterns are short patterns that consist of only fixed positions (positions with only one specified amino acid) and wildcard positions (positions with any amino acid allowed), with two fixed positions at the end of the patterns. Although these methods are successful in identifying certain SLiMs, many of them do not allow flexible gaps in SLiMs [18, 20, 21], or demand additional information besides sequences [16, 17], which prevent them from general applications. In addition, the few methods [15, 19] that can be applied for general SLiM discovery do not work well with large datasets, according to a recent study [18]. Finally, the significance assessment of peptide patterns is still problematic in current methods [1, 23]. For instance, the occurrence number of a SLiM in a sequence is often assumed to follow a binomial distribution [15, 23]. However, due to the dependency of adjacent overlapping segments in the sequence, this assumption is violated. Without a reliable approach for significance evaluation, available methods are not effective to identify SLiMs in noisy data.

Here we propose a novel approach, FlexSLiM, for de novo SLiM discovery in a group of protein sequences. Starting from selected dipeptide elementary patterns, FlexSLiM groups these dipeptides

ICBCB 2018, March 12–14, 2018, Chengdu, China

ACM ISBN 978-1-4503-6348-8/18/03…$15.00
https://doi.org/10.1145/3194480.3194501

such that each group of dipeptides is likely contained in one and only one SLiM. FlexSLiM then combine the dipeptides in each group into potential SLiMs that may contain flexible gaps and degenerate positions and assess their significance. Our study shows that FlexSLiM is capable of discovering SLiMs from both small sets of protein sequences, and massive proteome-wide observations. In addition, it requires no additional information other than protein sequences, which enables its general applications. Finally, FlexSLiM can accurately calculate statistical significance of SLiMs. Given the fact that hundreds of SLiMs have still to be discovered [1, 23], our study provides a powerful tool that will significantly advance our understanding of SLiMs and SLiM functions.

2. METHODS

2.1 Representation of SLiMs

SLiMs often contain a mixture of defined positions and undefined positions [23]. A defined position can be called either a fixed position when only one specific amino acid is allowed in this position or a degenerate position when several specified amino acids could be in this position. In contrast, undefined positions are often wildcard positions where any of the 20 amino acids is allowed. Because of the existence of wildcard positions, a SLiM often contains flexible gaps that are stretches of a variable number of consecutive wildcard positions. For instance, the SLiM Fun_Delta [1], represented by the regular expression [DE].{2,4}NN[IL], contains a flexible gap of 2 to 4 amino acids long that starts at the second position. A regular expression is a concise means to describe groups of strings and regular expressions are commonly used to describe SLiMs [1]. This above regular expression, [DE].{2,4}NN[IL], means that the first position of the Fun_Delta SLiM is a degenerate position with D or E, then followed by this flexible gap, two fixed positions with N, and a degenerate position with either I or L.

2.2 Framework of FlexSLiM

Effective methods are urgently needed to identify SLiMs. However, it is challenging to directly identify SLiMs, because a SLiM may include a huge number of possible peptide patterns. For instance, the Fun_Delta SLiM [DE].{2,4}NN[IL] mentioned above contains $2 \times (20^2 + 20^3 + 20^4) \times 1 \times 1 \times 2 = 673{,}600$ different peptide patterns with only fixed positions. We designate the peptide patterns contained in a SLiM as SLiM induced patterns, such as DRCNNI and D..NNI for the Fun_Delta SLiM. Because of the large number of induced patterns contained in a SLiM, many induced patterns are not statistically significant and it is thus also difficult to identify all induced patterns directly. To resolve the above issues, we propose to identify short elementary patterns first and then combine elementary patterns into SLiMs. However, the number of short elementary patterns could be still large and we cannot afford to check whether each pair of elementary patterns can be combined into longer peptides. An efficient approach for combining elementary patterns is needed. In addition, SLiMs are quite weak patterns due to the multiple choices of amino acids at a position and the flexible gaps. It is thus necessary to have an accurate way to assess the statistical significance of SLiMs. We propose a computational framework for SLiM identification by taking these factors into account. The proposed approach is composed of four steps detailed below: (1) select dipeptide elementary patterns; (2) generate SLiM induced patterns; (3) Discover SLiMs; and (4) calculate significance of patterns.

2.3 Select Dipeptide Elementary Patterns

To select elementary patterns that are substrings of SLiMs to be identified, previous methods often require the occurrence number of elementary patterns be larger than a predefined cutoff [15, 20]. However, elementary patterns in a SLiM have different occurrence probabilities and thus have different distributions of occurrence numbers in input sequences. For example, for the SLiM LIG_Rb_LxCxE_1, [LI].C.[DE], the occurrence number of the dipeptide patterns it contains, L.C.E, L.C.D, I.C.E and I.C.D, is 28, 3, 1, and 0, respectively, in the 32 experimentally validated instances in the ELM database [1]. Therefore, it is difficult to set a predefined cutoff properly. If the cutoff is small, many unrelated elementary patterns are selected, which increases the computation time substantially and the difficulty in pinpointing true SLiMs. If the cutoff is large, useful elementary patterns with low occurrence, such as I.C.D and L.C.E, are removed.

To solve the problem, we distinguish two types of elementary patterns: backbone elementary patterns and branch elementary patterns. Backbone elementary patterns are those with the occurrence number larger than a predefined occurrence cutoff, OC. Branch patterns are those with the occurrence number smaller than the occurrence cutoff OC while having its z-score larger than a predefined z-score cutoff, ZC. In this paper, the z-score of a pattern is defined as its occurrence number minus its expectation divided by its standard deviation.

For a given group of sequences, first, we obtain all dipeptides in these sequences. A dipeptide is a peptide segment like x.{n}y, where (i) x and y are two amino acids; (ii) n represents the number of wildcard positions between x and y. Second, we obtain the occurrence number of all dipeptides and select backbone dipeptides by using the occurrence cutoff OC. Third, we obtain the frequency of 20 amino acids in input sequences, and calculate z-scores of non-backbone dipeptides in input sequences. Finally, we select branch dipeptides that have a z-score larger than the z-score cutoff ZC.

2.4 Generate SLiM Induced Patterns

With elementary patterns, a common approach to obtain SLiM induced patterns is to combine similar elementary patterns into longer elementary patterns [15, 19]. For instance, starting from dipeptide elementary patterns, this strategy will extend dipeptides into 3-mers, and then further extend 3-mers into 4-mers. This process is repeated until the number of fixed positions in the patterns reaching a pre-defined maximum value. Every time when a pattern will be extended, this strategy needs to compare this pattern with other patterns, which is time-consuming. In addition, after extension, this strategy needs to determine whether the extended patterns occur in enough sequences, which takes a lot of time as well.

To make this process more efficient, we group elementary patterns first such that each group of elementary patterns is likely from a SLiM. In this way, we only need to compare a much smaller number of patterns within a group in order to extend a pattern every time. In addition, we do not need to check whether the extended patterns occur in enough sequences because our grouping algorithms guarantee that a group of elementary patterns occur in enough sequences. The details of how to obtain induced patterns are as follows (Figure 1).

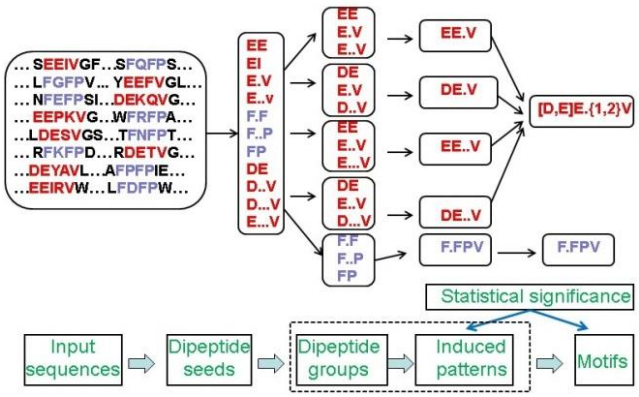

Figure 1. The pipeline in FlexSLiM.

First, we apply the frequent pattern mining algorithms [24, 25] to group backbone elementary patterns. The typical problem the frequent pattern mining algorithms can address is: what products (apple, orange, juice, onion, etc) do customers frequently buy together, given a database that stores the products each customer bought at a time? If we treat backbone dipeptides as a type of products and each sequence as a customer, these algorithms could identify groups of backbone dipeptides whose occurrence numbers are greater than a pre-defined cutoff. We call this cutoff the support cutoff, abbreviated as SC. Note that the elementary patterns contained in a SLiM should occur within windows of no longer than 11 amino acids long in many sequences [23] while the frequent pattern mining algorithms do not take the relative locations of dipeptides in input sequences into account. However, since the average length of human proteins is about 428 amino acid long based on all human proteins in the Ensembl database [26], it is rare that multiple non-overlapping dipeptides co-occur in multiple sequences, especially if we remove segments such as globular domains that are unlikely to contain SLiM instances [27]. So the majority of extended patterns satisfy the support cutoff, SC.

Second, we insert similar branch dipeptides into every group of backbone dipeptides. For a given group of backbone dipeptides, we will calculate the similarity of each backbone dipeptide in this group with every branch dipeptide. The similarity of two dipeptides is defined as the sum of the substitution scores of the two pairs of amino acids in the two dipeptides. For instance, the similarity of two dipeptides, A..B and B.C, is defined as the sum of the substitution scores of (A, B) and (B, C). The BLOSUM62 substitution matrix is used to calculate the substitution score of each pair of amino acids. If the similarity of two dipeptides is greater than or equal to -1, we claim this pair of dipeptides similar. We select -1 as the cutoff to define similar patterns because we checked all SLiMs in the ELM database and found that more than 90% of dipeptides from two defined positions of the same SLiM in the ELM database have a similarity no small than -1.

Third, with groups of expanded dipeptides, we extend the dipeptides in the same group into longer patterns (Figure 1). These longer patterns are called SLiM induced patterns, which contain only wildcard positions and fixed positions. We will first obtain induced backbone patterns using only backbone dipeptides. If the last n-1 fixed positions of an n-mer match the first n-1 fixed positions of another n-mer, where an n-mer is a backbone dipeptide or an induced backbone pattern, the first n-mer can be extended with the last fixed position of the second n-mer. For example, A..B and B.C can be combined into A..B.C. The rationale behind this tail-head concatenation operation is that if A..B.C occurs more than the SC times, then the occurrence times of A..B and B.C is larger than SC as well. Besides extending backbone elementary patterns into induced backbone patterns, we will also obtain induced branch patterns by using one branch pattern and one backbone pattern. In detail, we will compare each backbone dipeptide with each branch dipeptide in the same group to see whether one matches the other except at one position. For instance, except at the second fixed position, A..B matches A.C. We will extend each pair of matched backbone and branch dipeptides into 3-mers. These 3-mers with z-scores larger than ZC are called branch 3-mers. Next, every branch 3-mer compared with every backbone 3-mer generated from the same group of dipeptides to obtain branch 4-mers. We will repeat this process until no pattern can be extended or the length of the extended pattern is larger than a length cutoff, L. In practice, this step of obtaining longer patterns takes little time, because the number of backbone dipeptides in the same group is small and the average number of defined positions in a SLiM is 3.7 and the average length of a SLiM is 6 based on the ELM database [23].

2.5 Discover SLiMs

We will implement the following procedure to construct SLiMs from SLiM induced patterns. First, we will transform the regular expression represented induced patterns into numbered patterns. In other word, to represent induced patterns, we will use the number of wildcard positions between two adjacent fixed positions in induced patterns to replace the wildcard positions. For instance, A..CD will be represented as A2C0D. With the numbered representation, the induced patterns for the same SLiM will have the same length. Moreover, the two types of flexibility in SLiMs, degenerate positions and flexible gaps, can be considered similarly after the conversion. Second, we will group numbered patterns with same length into the same group. Third, for each group of induced patterns, we will discover SLiMs by merging proper patterns, including backbone and branch patterns, into SLiMs. In detail, all backbone patterns in this group are arbitrarily ordered. Starting from i-th backbone pattern, we will find similar patterns that have an order larger than i. Two patterns are called similar only if they are exactly the same except one mismatch. That is, two similar patterns have either different amino acids or different gap sizes at one and only one position. Next, we sort all similar patterns for the i-th backbone pattern. Compared with the i-th backbone pattern, patterns with different amino acids are sorted at the beginning while with different gap sizes are sorted at the end. In addition, compared with the sequences containing the i-th backbone pattern, patterns occurring in different sequences will be sorted before patterns occurring in similar sequences. Next, we will keep combining the backbone pattern with the top similar patterns until the sequences containing the combined pattern is not increased or all similar patterns have been considered. The significance of a pattern is measured by a z-score, which is calculated by the method detailed in the following section. If the z-score of the combined pattern is smaller than a predefined cutoff, this combined pattern will be output. Finally, we will repeat this process for all i and all different lengths of induced patterns.

2.6 Calculate Significance of Patterns

We will first calculate the probability that a pattern occurs in a random sequence of length n. A pattern here can be a dipeptide, an induced pattern, or a SLiM. Since a pattern here can be represented by a regular expression and a regular expression can be described by a *deterministic finite automaton (DFA)*, We can describe a pattern by a *DFA* [28]. With the DFA corresponding to

a pattern, any random sequence of length n can be thought as a path in a graph representing this DFA [29, 30]. Similar to previous studies in DNA sequences [30], we can calculate the probability that this pattern occurs in a protein sequence of length n.

We will then calculate the significance of this pattern in multiple protein sequences. Assume we have m sequences, the length of which are $n_1, \cdots, n_m$. Assume a pattern occurs in m_0 out of the m sequences. How significant is this pattern? To address this problem, we will first calculate p_i, the probability that this pattern occurs in the a random sequence of n_i amino acid long, by the above DFA based method [29, 30]. The z-score of this pattern is defined as:

$$\frac{\boldsymbol{m_0 - \sum_{i=1}^{m} p_i}}{\boldsymbol{\sum_{i=1}^{m} p_i(1 - p_i)}}$$

3. RESULTS

3.1 FlexSLim, a Novel Approach for SLiM Discovery

We developed a novel approach, FlexSLiM, for SLiM discovery in protein sequences. In brief, given a group of input sequences, first, we grouped protein sequences into different Unrelated Protein Clusters (UPCs) by using bl2seq in BLAST software package. Second, masked ordered regions and low complexity regions by using IUPRED [31] and SEG [32]. Third, FlexSLiM selects backbone 2-mers with large occurrence numbers and backbone 2-mers with low occurrence numbers while high z-scores. Fourth, FlexSLiM groups the backbone 2-mers such that a group of backbone 2-mers is likely from the same SLiM. Fifth, FlexSLiM inserts similar branch 2-mers into each group of backbone 2-mers. Sixth, FlexSLiM constructs longer backbone patterns from groups of backbone 2-mers and longer branch patterns by combining backbone and branch elementary patterns separately to obtain induced patterns. Finally, FlexSLiM combine similar induced patterns into SLiMs, which may have degenerate positions and flexible gaps.

Compared with available methods for SLiM discovery [15, 18-20], FlexSLiM have several unique features. First, FlexSLiM uses double cutoffs to select elementary patterns. This is superior to the strategy of using a simple occurrence cutoff, because frequent elementary patterns contained in SLiMs are found by the occurrence cutoff efficiently, while infrequent elementary patterns are detected as many as possible by the z-score cutoff. This advantage of the double cutoffs is even more evident when there are a large number of input sequences, as the occurrence cutoff can significantly reduce the number of elementary patterns that needs to be considered and the z-score cutoff can dramatically improve the accuracy of including essential elementary patterns. Second, FlexSLiM applies the frequent pattern mining algorithms [24, 25] to group backbone elementary patterns and then inserts similar branch elementary patterns into each group. In this way, FlexSLiM only needs to compare a small number of patterns each time when it extends patterns and does not need to verify whether the extended patterns occur in enough sequences. This is different from the common approach used by other methods [15, 19, 20], in that other methods often have to compare a large number of elementary patterns to determine which pairs of elementary patterns to be combined to generate longer patterns and have to check whether the extended patterns occur in many sequences. Third, FlexSLiM proposes an accurate way to calculate the significance of SLiMs. Previous methods often assume a binomial distribution [15] to calculate the probability that a pattern occurs in one or multiple sequences, which neglects the complexity of the induced patterns in SLiMs and the overlap of SLiM instances in sequences. Because of these unique features, FlexSLiM performs well on the following simulated data and experimental data.

3.2 Studies on Simulated Data

We tested FlexSLiM on simulated datasets. First, we chose all SLiMs from the benchmark used by SLiMFinder [15]. Next, we generated random sequences based on two different sets of amino acid frequencies and length distributions. One set adopts length range distribution [131, 331], whose mean is the average length of unmasked regions in all benchmark motifs, and the amino acid frequencies in unmasked regions for the inserted motif. The other set adopts length range distribution [254, 454], whose mean is the average length of all protein sequences, and the amino acids frequencies in both masked and unmasked regions for the inserted motif. In each experiment, 100 random sequences were generated and 50 of them were inserted with SLiM instances of the corresponding SLiM. 20 of 50 are the same instance, which is taken as backbone, and 30 of 50 are randomly picked from all possible instances. The backbone instances are generated by using first letter at defined positions and shortest length of flexible gaps. Finally, we applied FlexSLiM to these datasets and compared with the SLiMFinder. We did not compare with DiLiMOT, since it cannot handle the 100 sequences used in the experiments. The parameters for FlexSLiM were L=8, G=4, N=5, M=4, OC=1/5*(number of cluster), ZC=1, and SC=1/3*(number of clusters), where L is the maximum length of SLiMs, G is the maximum size of flexible gaps, N is the maximum number of defined positions in a SLiM, and M is the maximum number of amino acids in one defined position. The default parameters were used for SLiMFinder. And the mask option is set to off for both softwares.

Form Table 1, we can see that both SLiMFinder and FlexSLiM did well in all 17 simulated datasets. FlexSLiM try to discovery large support patterns firstly. Therefore small support but high significant patterns are not considered as candidates. For example, SLiMFinder finds out large significant patterns HDEL and QQEL, both of which are instances of inserted motifs of TRG_ER_KDEL_1. On the other hand, only KDEL is taken as backbone by FlexSLiM. As a result, only KDEL is expanded into two predictions. But just expanding backbones may cause losing defined positions. For example FlexSLiM find less defined positions than SLiMFinder for motif LIG_AP_GAE_1. And from the table, we can see that both software find better results by using unmasked regions. This implies masking is indispensible strategy for slim discovery.

The effect of finding SLiMs without a predefined equivalency list is evident in Table 1. SLiMFinder uses a predefined equivalency list to determine the set of amino acids allowed at a degenerate position, while FlexSLiM does not use such a list. Without the equivalency list, FlexSLiM can obtain more flexible patterns. For example, {K,S} is not in any equivalency lists. Therefore SLiMFinder cannot discovery [KS][DEN]EL for motif TRG_ER_KDEL_1.

Table 1. Tests on random sequences with different lengths and amino acids frequencies

ELM Accession	Regular Expression	FlexSLiM		SLiMFinder	
		231	354	231	354
TRG_ER_KDEL_1	[KRHQSAP][DENQT]EL	[KS][DEN]EL	[KQS][DEQ]EL	HDEL	QQEL
LIG_Dynein_DLC8_1	[KR].TQT	[KR].TQT	[KR].TQT	[KR].TQT	[KR].TQT
LIG_PCNA	Q..[ILM]..[DFHM][FMY]	Q..[IL]…F	Q..I..D[FMY]	Q..I.CHY	Q.E.H.MY
MOD_SUMO	[VILMAFP]K.E	[VLMA]K.E	[VIF]K.E	VKTE	IKSE
LIG_SH3_2	P..P.[KR]	P..P.[KRQ]	P..P.K	PRD.GK	PEH.SR
LIG_CYCLIN_1	[RK].L.{0,1}[FYLIVMP]	[RK].L.F	[RK].L.F	KSLSM	KSLL
LIG_CtBP	[PG][LVIPME][DENS]L[VASTRGE]	PL[DEN]V	P.DLV	PLNLR	GPELV
LIG_AP_GAE_1	[DE][DES][DEGAS]F[SGAD][DEAP][LVIMFD]	[DE]D..S.L	D..F[GS][DE]	[DE]DDF..L	D..F[GS][DE]L
LIG_14-3-3_3	[RHK][STALV].[ST].[PESRDIFTQ]	R..[ST].P	RS.[ST]	RLQ.PQ	RSE.MR
LIG_Rb_LxCxE_1	[LI].C.[DE]	[LI].C.[DE]	[LI].C.[DE]	LNCND	LLCLE
LIG_Clathr_ClatBox_1	L[IVLMF].[IVLMF][DE]	L[IV].V[DE]	L[IM].[VML][DE]	LVHVD	LMQMD
LIG_14-3-3_1	R..[ST].P	R..[ST].P	R..S.P	RGAT.P	R.NSQP
LIG_RGD	RGD	RGD	RGD	RGD	RGD
LIG_HP1_1	P[MVLIRWY]V[MVLIAS][LM]	P[LMY]V[MIS]	PMV[MIS]M	P[LM]V[IM][LM]	PWVSM
LIG_NRBOX	L..LL	L..LL	L..LL	LKFLL	LTTLL
MOD_N-GLC_2	N.C	N.C	N.C	NFC	NPC
TRG_LysEnd_APsAcLL_1	[DERQ]…L[LVI]	[DR]…L[LI]	D...L[LV]	EA.RLI	RKYD.I

The top one predictions are shown in the table. The two numbers, 231 and 351, are the average length of the simulated sequences.

3.3 Studies on Benchmark Data

The motif benchmark datasets tested by SLiMFinder were used in our experiment. Two other tools, SLiMFinder and DiLiMOT were also used to compare with FlexSLiM. Because the ELM database has been updated, we redo all the tests on both SLiMFinder and DiLiMOT. The corresponding protein sequences that contain each of the SLiMs were obtained from the ELM database [33]. The parameters for the two tools were set to be the default parameters.

The parameters for FlexSLiM were the same with those in the simulated experiment except OC=1/3 of clusters, SC=0.5 of clusters. If OC is set to 3 if it is less than 3. The results were shown in Table 2.

For the 17 SLiMs, FlexSLiM predicted similar SLiMs to each of them. In some cases, FlexSLiM does not provide similar prediction for the known motif. This is because the OC is too large or the backbone patterns have too short support. Then FlexSLiM cannot discover the proper backbones and therefore the expansion is invalid. We can see the selection of OC is very important for FlexSLiM. On the other hand, if there is not a large support backbone pattern in the input sequences, FlexSLiM would convert into similar mechanism as previous algorithms.

The advantage of using double cutoffs in FlexSLiM was demonstrated here. For the 4-mers contained in the SLiM TRG_ER_KDEL_1 in Table 2, HDEL, it only occurs once in the input sequences. SLiMFinder needs to set the occurrence cutoff to be one in order to select the dipeptide as elementary pattern. However, all elementary patterns would be considered if the occurrence cutoff is set so low, which would take a long time to extend the selected elementary patterns into SLiMs. By using the double cutoff strategy, the induced pattern can be detected precisely. In addition, although the total number of elementary patterns considered in by the three tools is the same, FlexSLiM does not consider the extension of branch elementary patterns themselves. In general, the percentage of branch elementary patterns in all elementary patterns is over 80%. As a result, the computation time of FlexSLiM is much shorter than that of SLiMFinder.

Table 2. Tests on benchmark datasets

ELM Accession	FlexSLiM		SLiMFinder		DiLiMOT	
	Result	Sig	Result	sig	Result	Sig
TRG_ER_KDEL_1 [KRHQSAP][DENQT]EL	[KH]DEL	1.66E-09	KDEL	0	KDEL	3.69E-40
LIG_Dynein_DLC8_1 [KR].TQT	S..K.TQT	1.32E-09	S..K.TQT	1.28E-05	K.TQT	6.18E-26
LIG_PCNA Q..[ILM]..[DFHM][FMY]	Q….S[FH]F	1.44E-15	Q.[ST][IL]..FF	4.43E-13	Q…..FF	0

MOD_SUMO [VILMAFP]K.E	S[AT].R	7.51E-03	[FIV]K.E	2.66E-06	VK.E(8)	1.55E-21
LIG_SH3_2 [P..P].[KR]	EIL.K	1.34E-08	EIL	7.80E-01	PP..P.R(4)	6.22E-21
LIG_CYCLIN_1 [RK].L.{0,1}[FYLIVMP]	[KR]R.L	5.64E-05	[KR][KR]..F	1.50E-02	YISP	1.38E-35
LIG_CtBP [PG][LVIPME][DENS][VASTRGE]	P.DL(2)	1.93E-03	P[ILM]DL	2.00E-03	P.DLS(5)	1.13E-47
LIG_AP_GAE_1 [DE][DES][DEGAS]F[SGAD][DEAP][VLIMFD]	D.FG.F(2)	1.14E-06	D.F..F.S..P	5.60E-01	DD.F..F	4.30E-64
LIG_14-3-3_3 [RHK][STALV].[ST].[PESRDIFTQ]	D.S.{2,4}S	2.27E-08	D..Y.D..PG	7.90E-02	R..S.D.S(2)	3.74E-29
LIG_Rb_LxCxE_1 [LI].C.[DE]	G..T	2.35E-01	L.C.E	2.12E-04	L.C.E	4.30E-48
LIG_Clathr_ClatBox_1 L[IVLMF].[IVLMF][DE]	LLDL(2)	3.81E-08	A..T..[FV]	1.00E-01	LL.LD(4)	3.46E-24
LIG_14-3-3_1 R..[ST].P	RS.S.P(5)	1.25E-04	T[FM].T	9.80E-01	RS.S.P	4.27E-25
LIG_RGD RGD	EE..A	4.37E-01	RGD	4.70E-01	RGD	9.55E-30
LIG_HP1_1 P[MVLIRWY]V[MVLIAS][LM]	R…D..R.G	5.45E-06	I..S..I[IM]	5.50E-02	VP.V.L(4)	8.92E-28
LIG_NRBOX L..LL	TGP.PG	3.45E-06	IK.E..D	8.40E-02	P.L.K	1.20E-26
MOD_N-GLC_2 N.C	RGDS	7.65E-06	G..WK	6.10E-01	EAP	2.35E-12
TRG_LysEnd_APsAcLL_1 [DERQ]…L[LVI]	[KP]K	8.62E-01	QE.V..[IV]	3.10E-01	E..R.L.F	1.37E-16

The top one predictions are shown in the table. If the top one prediction is not similar with the motif, the most similar top ranked prediction is shown in the table. And the corresponding rank is shown in the brackets.

4. DISCUSSION

We developed a novel approach, FlexSLiM, to discover general SLiMs in protein sequences. As previous methods [15, 19, 20], FlexSLiM starts from elementary patterns and then extends elementary patterns to SLiMs with degenerate positions and flexible gaps allowed. Different from previous methods, FlexSLiM distinguishes two types of elementary patterns and applies frequent pattern mining algorithms to extend elementary patterns, which greatly speed up the process of SLiM discovery and improve the accuracy of SLiM discovery. As shown in the simulated data and experimental data, FlexSLiM shows superior or comparable performance when compared with two state of the art best methods [15, 19].

FlexSLiM uses DFA to calculate the exact occurrence probability of a pattern in a random sequence. In some specific cases, the number of states in a DFA is large. For example, in the SLiM TRG_NLS_Bipartite_1 in ELM [1], there is a flexible gap .{7, 15}. As a result, the probability calculation based on DFA takes time. This is because the number of states of the DFA representing .*A.{m, n}B.* is O(k+2m-n+1) [34], where k is the number of different lengths of the pattern. Fortunately, for the majority of SLiMs, m-n+1 is a small number. In our implementation, if the number of DFA states exceeds 100, the program automatically choose to use approximate methods to compute the probability.

Like SLiMFinde, FlexSLiM masks domains and non-disordered regions in input sequences before finding SLiMs. Although the percent of SLiM instances existing in those regions is small, many instances will be masked for certain SLiMs. To resolve this issue, one could find SLiMs in masked sequences first and then try to detect the masked instances by using regular expression and automata theory [30]. This is not implemented in FlexSLiM and will be our future research subject. In addition, SLiMs with only two defined positions could not be discovered by FlexSLiM, since we use dipeptides as elementary patterns. To remedy this problem, we can consider dimer elementary patterns similarly as dipeptide elementary patterns and combine the results from dipeptide elementary patterns with those from the dimer patterns.

5. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation [1356524, 1149955, 1661414]; and the National Institutes of Health [R15GM123407].

6. REFERENCES

[1] H. Dinkel, K. Van Roey, S. Michael, M. Kumar, B. Uyar, B. Altenberg, V. Milchevskaya, M. Schneider, H. Kuhn, A. Behrendt, S. L. Dahl, V. Damerell, S. Diebel, S. Kalman, S. Klein, A. C. Knudsen, C. Mader, S. Merrill, A. Staudt, V. Thiel, L. Welti, N. E. Davey, F. Diella, and T. J. Gibson, "ELM 2016--data update and new functionality of the eukaryotic linear motif resource," *Nucleic Acids Res,* vol. 44, no. D1, pp. D294-300, Jan 4, 2016.

[2] S. Lemeer, and A. J. R. Heck, "The phosphoproteomics data explosion," *Current Opinion in Chemical Biology,* vol. 13, no. 4, pp. 414-420, Oct, 2009.

[3] S. P. Mirza, and M. Olivier, "Methods and approaches for the comprehensive characterization and quantification of cellular proteomes using mass spectrometry," *Physiological Genomics,* vol. 33, no. 1, pp. 3-11, Mar 14, 2008.

[4] H. Dumortier, J. K. Gunnewiek, J. P. Roussel, Y. van Aarssen, J. P. Briand, W. J. van Venrooij, and S. Muller, "At least three linear regions but not the zinc-finger domain of

U1C protein are exposed at the surface of the protein in solution and on the human spliceosomal U1 snRNP particle," *Nucleic Acids Research,* vol. 26, no. 23, pp. 5486-5491, Dec 1, 1998.

[5] M. Kikuchi, M. Kataoka, T. Kojima, T. Horibe, K. Fujieda, T. Kimura, and T. Tanaka, "Single chain antibodies that recognize the N-glycosylation site," *Archives of Biochemistry and Biophysics,* vol. 422, no. 2, pp. 221-229, Feb 15, 2004.

[6] S. Basu, and D. Plewczynski, "AMS 3.0: prediction of post-translational modifications," *BMC Bioinformatics,* vol. 11, pp. 210, Apr 28, 2010.

[7] R. Gutman, C. Berezin, R. Wollman, Y. Rosenberg, and N. Ben-Tal, "QuasiMotiFinder: protein annotation by searching for evolutionarily conserved motif-like patterns," *Nucleic Acids Research,* vol. 33, pp. W255-W261, Jul 1, 2005.

[8] T. Mi, J. C. Merlin, S. Deverasetty, M. R. Gryk, T. J. Bill, A. W. Brooks, L. Y. Lee, V. Rathnayake, C. A. Ross, D. P. Sargeant, C. L. Strong, P. Watts, S. Rajasekaran, and M. R. Schiller, "Minimotif Miner 3.0: database expansion and significantly improved reduction of false-positive predictions from consensus sequences," *Nucleic Acids Research,* vol. 40, no. D1, pp. D252-D260, Jan, 2012.

[9] J. C. Obenauer, L. C. Cantley, and M. B. Yaffe, "Scansite 2.0: proteome-wide prediction of cell signaling interactions using short sequence motifs," *Nucleic Acids Research,* vol. 31, no. 13, pp. 3635-3641, Jul 1, 2003.

[10] C. Ramu, "SIRW: a web server for the Simple Indexing and Retrieval System that combines sequence motif searches with keyword searches," *Nucleic Acids Research,* vol. 31, no. 13, pp. 3771-3774, Jul 1, 2003.

[11] E. Olorin, K. T. O'Brien, N. Palopoli, A. Perez-Bercoff, D. C. Shields, and R. J. Edwards, "SLiMScape 3.x: a Cytoscape 3 app for discovery of Short Linear Motifs in protein interaction networks," *F1000Res,* vol. 4, pp. 477, 2015.

[12] H. Y. K. Lam, P. M. Kim, J. Mok, R. Tonikian, S. S. Sidhu, B. E. Turk, M. Snyder, and M. B. Gerstein, "MOTIPS: Automated Motif Analysis for Predicting Targets of Modular Protein Domains," *Bmc Bioinformatics,* vol. 11, May 11, 2010.

[13] D. Schwartz, and S. P. Gygi, "An iterative statistical approach to the identification of protein phosphorylation motifs from large-scale data sets," *Nature Biotechnology,* vol. 23, no. 11, pp. 1391-1398, Nov, 2005.

[14] M. Fuxreiter, P. Tompa, and I. Simon, "Local structural disorder imparts plasticity on linear motifs," *Bioinformatics,* vol. 23, no. 8, pp. 950-6, Apr 15, 2007.

[15] R. J. Edwards, N. E. Davey, and D. C. Shields, "SLiMFinder: a probabilistic method for identifying over-represented, convergently evolved, short linear motifs in proteins," *PLoS One,* vol. 2, no. 10, pp. e967, Oct 3, 2007.

[16] J. Hu, and F. Zhang, "BayesMotif: de novo protein sorting motif discovery from impure datasets," *BMC Bioinformatics,* vol. 11 Suppl 1, pp. S66, Jan 18, 2010.

[17] W. Hugo, F. Song, Z. Aung, S. K. Ng, and W. K. Sung, "SLiM on Diet: finding short linear motifs on domain interaction interfaces in Protein Data Bank," *Bioinformatics,* vol. 26, no. 8, pp. 1036-42, Apr 15, 2010.

[18] D. S. Lieber, O. Elemento, and S. Tavazoie, "Large-Scale Discovery and Characterization of Protein Regulatory Motifs in Eukaryotes," *Plos One,* vol. 5, no. 12, Dec 29, 2010.

[19] V. Neduva, and R. B. Russell, "DILIMOT: discovery of linear motifs in proteins," *Nucleic Acids Research,* vol. 34, pp. W350-W355, Jul 1, 2006.

[20] I. Rigoutsos, and A. Floratos, "Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm (vol 14, pg 55, 1998)," *Bioinformatics,* vol. 14, no. 2, pp. 229-229, 1998.

[21] S. H. Tan, W. Hugo, W. K. Sung, and S. K. Ng, "A correlated motif approach for finding short linear motifs from protein interaction networks," *Bmc Bioinformatics,* vol. 7, Nov 16, 2006.

[22] N. E. Davey, J. L. Cowan, D. C. Shields, T. J. Gibson, M. J. Coldwell, and R. J. Edwards, "SLiMPrints: conservation-based discovery of functional motif fingerprints in intrinsically disordered protein regions," *Nucleic Acids Res,* vol. 40, no. 21, pp. 10628-41, Nov, 2012.

[23] N. E. Davey, K. Van Roey, R. J. Weatheritt, G. Toedt, B. Uyar, B. Altenberg, A. Budd, F. Diella, H. Dinkel, and T. J. Gibson, "Attributes of short linear motifs," *Mol Biosyst,* vol. 8, no. 1, pp. 268-81, Jan, 2012.

[24] G. Grahne, and J. F. Zhu, "Fast algorithms for frequent itemset mining using FP-trees," *Ieee Transactions on Knowledge and Data Engineering,* vol. 17, no. 10, pp. 1347-1362, Oct, 2005.

[25] J. W. Han, J. Pei, and Y. W. Yin, "Mining frequent patterns without candidate generation," *Sigmod Record,* vol. 29, no. 2, pp. 1-12, Jun, 2000.

[26] D. R. Zerbino, P. Achuthan, W. Akanni, M. R. Amode, D. Barrell, J. Bhai, K. Billis, C. Cummins, A. Gall, C. G. Giron, L. Gil, L. Gordon, L. Haggerty, E. Haskell, T. Hourlier, O. G. Izuogu, S. H. Janacek, T. Juettemann, J. K. To, M. R. Laird, I. Lavidas, Z. Liu, J. E. Loveland, T. Maurel, W. McLaren, B. Moore, J. Mudge, D. N. Murphy, V. Newman, M. Nuhn, D. Ogeh, C. K. Ong, A. Parker, M. Patricio, H. S. Riat, H. Schuilenburg, D. Sheppard, H. Sparrow, K. Taylor, A. Thormann, A. Vullo, B. Walts, A. Zadissa, A. Frankish, S. E. Hunt, M. Kostadima, N. Langridge, F. J. Martin, M. Muffato, E. Perry, M. Ruffier, D. M. Staines, S. J. Trevanion, B. L. Aken, F. Cunningham, A. Yates, and P. Flicek, "Ensembl 2018," *Nucleic Acids Res,* vol. 46, no. D1, pp. D754-D761, Jan 4, 2018.

[27] M. Glittenberg, C. Pitsouli, C. Garvey, C. Delidakis, and S. Bray, "Role of conserved intracellular motifs in Serrate signalling, cis-inhibition and endocytosis," *EMBO J,* vol. 25, no. 20, pp. 4697-706, Oct 18, 2006.

[28] J. E. Hopcroft, and J. D. Ullman, *Introduction to automata theory, languages, and computation*, Reading, Mass.: Addison-Wesley, 1979.

[29] G. Nuel, L. Regad, J. Martin, and A. C. Camproux, "Exact distribution of a pattern in a set of random sequences generated by a Markov source: applications to biological data," *Algorithms for Molecular Biology,* vol. 5, Jan 26, 2010.

[30] P. Ribeca, and E. Raineri, "Faster exact Markovian probability functions for motif occurrences: a DFA-only approach," *Bioinformatics,* vol. 24, no. 24, pp. 2839-2848, Dec 15, 2008.

[31] Z. Dosztanyi, V. Csizmok, P. Tompa, and I. Simon, “IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content,” *Bioinformatics,* vol. 21, no. 16, pp. 3433-4, Aug 15, 2005.

[32] J. C. Wootton, and S. Federhen, “Statistics of Local Complexity in Amino-Acid-Sequences and Sequence Databases,” *Computers & Chemistry,* vol. 17, no. 2, pp. 149-163, Jun, 1993.

[33] C. M. Gould, F. Diella, A. Via, P. Puntervoll, C. Gemund, S. Chabanis-Davidson, S. Michael, A. Sayadi, J. C. Bryne, C. Chica, M. Seiler, N. E. Davey, N. Haslam, R. J. Weatheritt, A. Budd, T. Hughes, J. Pas, L. Rychlewski, G. Trave, R. Aasland, M. Helmer-Citterich, R. Linding, and T. J. Gibson, “ELM: the status of the 2010 eukaryotic linear motif resource,” *Nucleic Acids Res,* vol. 38, no. Database issue, pp. D167-80, Jan, 2010.

[34] F. Yu, Z. Chen, Y. Diao, T. V. Lakshman, and R. H. Katz, "Fast and memory-efficient regular expression matching for deep packet inspection."