Indian Journal of Science and Technology, *Vol 10(25), DOI: 10.17485/ijst/2017/v10i25/101419, July 2017*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

ProMot: A Tool for the Fast Discovery of Functional Motifs from Aligned Protein Sequences

Akash Nag* **and Sunil Karforma**

Department of Computer Science, The University of Burdwan, Rajbati Burdwan – 713104, West Bengal, India;
nag.akash.cs@gmail.com, sunilkarforma@yahoo.com

Abstract

Objectives: We present an algorithm to quickly identify conserved patterns from a set of aligned protein sequences. **Method:** Using contribution statistics, the proposed method identifies a motif describing the given set of sequences, and it is flexible enough to identify variable-length wildcard regions and also identifying motif elements based on regions containing amino-acids having similar physiochemical properties. In this paper, we compare its performance against other well-known motif-discovery algorithms, on three datasets: snake-toxins, insulin proteins, and methylated-DNA protein-cysteine methyl transferase active-site enzymes. **Findings:** When tested with 91 neurotoxin protein sequences from 45 species of Elapid snakes, the algorithm successfully generated a motif which had a 97% precision. The motif generated by our algorithm had a 92% precision on the Insulin family and 96.5% on the MGMT family of proteins. **Novelty:** Our algorithm is fast, efficient, outperforms on average the commonly used motif generation algorithms in terms of accuracy, and never fails to report any motifs unlike some other algorithms.

Keywords: Flexible Wildcard Regions, Insulin, Motifs, Motif Generation, Patterns, PROSITE, Protein Families, Snake Toxins

1. Introduction

A protein sequence motif of a protein family is a pattern-sequence of amino-acids that is conserved in most proteins of that protein-family, and is thought to be biologically significant for those proteins in exhibiting their structure or function. A protein family motif therefore can be said to be a representative pattern of that protein family, and most proteins in the said family shall conform to this pattern. The PROSITE[1] database is a database consisting of manually curated motifs, identifying various protein families. The motifs help in the identification of short well-conserved regions of proteins from various species. Such conserved regions may be the result of a shared phylogenetic relationship among the species. But more importantly, such conserved regions often serve either to provide a support to the structure of the protein, or to serve as functionally important parts of the protein. Discovery of such motifs then, is a precursor to better understanding the tertiary structure of a protein, and also to predict the function of that protein in those species.

In this paper, we propose a new algorithm, hereinafter called ProMot, which can efficiently and automatically generate motifs given a multiple sequence alignment of related protein sequences. The usually tedious process of manually generating motifs for protein families can then be automated using the proposed algorithm. Armed with such motifs, we can then recover other related proteins from protein databases that are similar to those in a given protein family. Such motifs can be useful in uncovering the signature of a protein family, which can then be used to quickly identify whether a new protein belongs to a specific family or not. By analyzing the strength of a motif, we can also estimate whether or not the concerned protein family is concrete or whether it is heterogenous and its constituent proteins need to be further sub-divided into sub-families. In other words, if a strong well conserved region is reflected in the generated motif, we can say with a high degree of confidence that the proteins belong to a single family. The motif strength can be gauged using the number of false-positives and false-negatives generated when that motif is used for searching against a protein database.

** Author for correspondence*

There are several existing algorithms for identification of motifs from protein sequences. Most of the algorithms[2–4] in this field work by dividing regions into *k*-tuples. Identification of conserved regions happens in groups of *k*-amino acids, and longer regions are found by concatenation of adjacent groups. Longer regions may be interspersed by variable-length wildcard regions allowing for any amino-acid to be present. Koza and Andre[5] developed an algorithm based on genetic programming, while Smith et al.[6] used fixed-spacing between adjacent groups of conserved blocks. The latter work was important in the sense that it provided the initial conserved segments to the BLOCKS database[7]. The primary disadvantage of most algorithms is that they provide some or other constraints on the motifs, with users having to specify either the minimum and maximum length of the motif, or the number of motif components, or the number of variable spacing, etc. Putting these restrictions allow the algorithms to execute faster, but presume that the user already has some knowledge of the structure of patterns present in the sequences. In terms of spacing lengths, variable-spacing is biologically more meaningful than fixed spacing because a number of insertion-deletion events may have happened during the evolution of the species from a common ancestor, and therefore, gap lengths may not be uniform among two related species between adjacent conserved regions.

A popular algorithm in this field is PRATT[8]. The primary advantage of this algorithm over our proposed one is that it does not require aligned sequences; however as we shall see in Section 4, our proposed method is better at identifying conserved patterns. HMMER[9] and MEME[10] are two other very widely used algorithms in this field whose results we also compare in the next section. The HMMER algorithm does not generate a motif in the sense of a consensus pattern of amino-acids, but produces a profile based on the hidden Markov model. This profile can then be used to search any protein database. Like our algorithm, it also requires a multiple sequence alignment to start with. A multiple sequence alignment of a set of protein sequences is a way of arranging those proteins in a way so as to identify regions of similarity between them, which is widely believed to be the result of some evolutionary relationship between the said proteins. MEME is another tool for motif discovery, and it does not require an aligned set of sequences as well; however, it can only produce ungapped fixed-length motifs, and hence is mostly unsuitable for discovering complex patterns. Another algorithm, DRIMust[11] also generates motifs from unaligned sequences; however it is sensitive to the order of the input sequences and finds sequences that are over-represented in the sequences that appear first in the input. Another disadvantage of the DRIMust web server interface[12] is that it cannot produce motifs of length greater than 20. A brief summary of these four algorithms are presented in Table 1.

In this paper, we compare our proposed algorithm against all four of these, as well as against the standard PROSITE motif, for each protein family under consideration. Among the four algorithms: HMMER is not a motif generation algorithm per-se but it is used for searching databases for homologs. However, our goal of developing ProMot is not only to develop sequence motifs but to ultimately search protein databases using those generated motifs to find all proteins in a given family. In doing so, both HMMER and ProMot can be said to be contributing to the same purpose. As far as DRIMust is concerned, it can only work when the input sequences are ranked, and it returns motifs that are over-represented in those sequences which are at the top. To maintain comparability with our algorithm, the sequences which remained mostly undetected (i.e. false

Table 1. Summary of the algorithms compared against our proposed algorithm

Algorithm	Strength	Weakness	Methodology
Pratt	Can work with both aligned and unaligned sequences	Precision not as good	Uses enumeration of triplets consisting of conserved residues with variable spacing between the residues
HMMER	Good precision, excellent sample fitness	Does not generate patterns; instead builds profiles	Uses hidden Markov models to build profiles
MEME	High precision	Cannot produce gapped motifs	Uses expectation-maximization by iteratively fitting a mixture-model to the sample
DRIMust	-	Unsatisfactory precision, motif length limited to 20	Uses suffix trees for enumeration of k-mers, which are assessed using mHG statistics

negatives) by the algorithms compared in this paper, were put at the end of the sequence lists before being presented to DRIMust so that it gives priority to those sequences which are detected by almost all motifs for that family. Regarding those proteins containing multiple conserved regions separated by unconserved regions, ProMot is well capable of handling them as well, where MEME fails as it only finds ungapped motifs. For the purposes of aligning the sequences whenever required, we used MUSCLE[13] because it is quite fast.

In Section 2, we discuss the proposed algorithm for motif generation along with the various parameters that can be used to tune its performance. In Section 3, we discuss about the sample data that have been used to test our algorithm. In Section 4, the results of motif generation using the sample data with our proposed algorithm as well as some commonly used algorithms have been discussed. In Section 5, we compare and contrast the quality of the motifs generated by the various algorithms. We also discuss the runtime performance of our algorithm and its software availability. Finally we conclude with our findings in Section 6.

2. Method: The Proposed Algorithm

2.1 User Parameters

The proposed algorithm, hereinafter called ProMot, takes three user specified parameters to search for motifs. Besides these, it also uses two constants in the algorithm. All of these are listed in Table 2. The FITNESS_THRESHOLD is the minimum required percentage of sample coverage (see Eqn. (1)). The MAX_CHOICES is the maximum number of allowed amino-acid choices in each motif-element.

$$Sample\ \ Coverage = \frac{H}{N} \times 100\% \qquad (1)$$

Where, H is the number of motif hits in the sample; and N is the sample size (number of sequences in the sample)>

2.2 The Method

The proposed algorithm is outlined in Figure 1. The input to the algorithm is a list of aligned protein sequences. The first step is determining a group for each column. The amino-acid choices for each group are the number of distinct amino-acids present in that column. The *min* and *max* values for a group are the minimum length of the group present in the motif; both are initially set to 1. Consecutive groups are merged by forming the union of the amino-acid choices, updating their contributions (probability of occurrence, see Eqn. (2)), and their *min* and *max* values appropriately. Gaps in the alignment result in setting *min* to 0. After the initial grouping, the algorithm enters the second phase, wherein for each group, the standard deviation (σ) is calculated (see Eqn. (3)). Groups are treated differently based on their σ values. If σ is more than the SD_THRESHOLD, the choice-list for that group are replaced with the complement of the choice-list, but only if it is shorter in length than the original list. If not, the group is replaced by a wildcard region ("x" in the motif) but only if the all the amino-acids in the choice-list are not contained in the same functional group according to the two of the classification types possible for proteins, as listed in Table 3.

On the other hand, if σ is more than SD_THRESHOLD, the choice-list is arranged in ascending order of their contributions. The cumulative sum of the normalized contributions of each amino-acid in the choice-list, are then computed using Eqn. (6).

Table 2. User Parameters and Constants

Parameters	Range/Values	Default	Meaning
FITNESS_THRESHOLD	1-100	-	The minimum percentage of the sequences that the motif must match
MAX_CHOICES	1-22	4	The maximum no. of dissimilar amino-acid choices per position allowed
ALLOW_GROUPING	Y / N	Y	Whether to check for grouping of amino-acids based on similar physio-chemical properties
Constants	**Value**		**Meaning**
SD_THRESHOLD	0.1		The cutoff Standard Deviation value determining the group treatment
CON_NORM_THRESHOLD	0.001		The cutoff Normalized Contribution value below which amino-acids are removed from a group

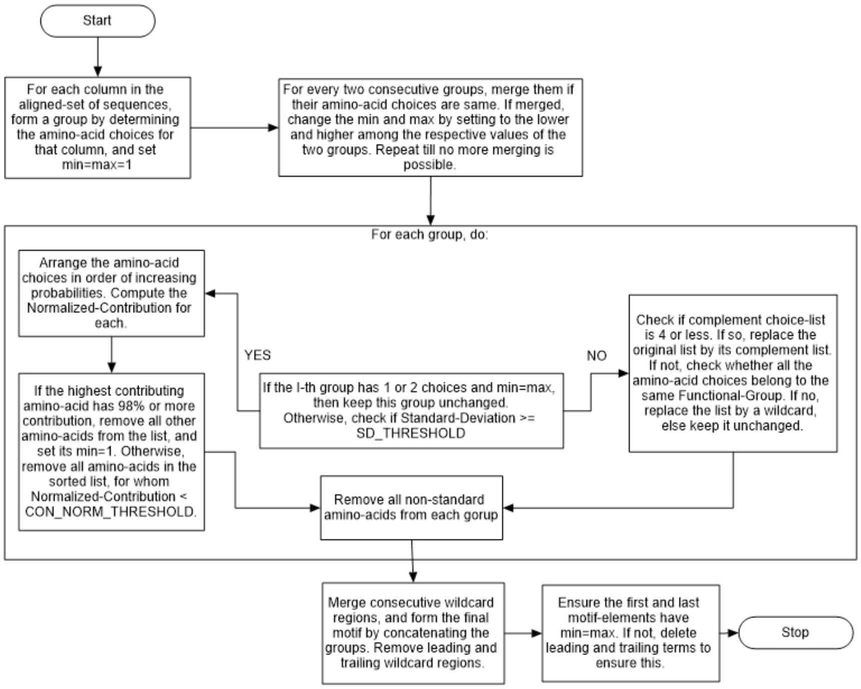
Start
For each column in the aligned-set of sequences, form a group by determining the amino-acid choices for that column, and set min=max=1
For every two consecutive groups, merge them if their amino-acid choices are same. If merged, change the min and max by setting to the lower and higher among the respective values of the two groups. Repeat till no more merging is possible.
For each group, do:
Arrange the amino-acid choices in order of increasing probabilities. Compute the Normalized-Contribution for each.
YES
NO
If the I-th group has 1 or 2 choices and min=max, then keep this group unchanged. Otherwise, check if Standard-Deviation >= SD_THRESHOLD
Check if complement choice-list is 4 or less. If so, replace the original list by its complement list. If not, check whether all the amino-acid choices belong to the same Functional-Group. If no, replace the list by a wildcard, else keep it unchanged.
If the highest contributing amino-acid has 98% or more contribution, remove all other amino-acids from the list, and set its min=1. Otherwise, remove all amino-acids in the sorted list, for whom Normalized-Contribution < CON_NORM_THRESHOLD.
Remove all non-standard amino-acids from each gorup
Merge consecutive wildcard regions, and form the final motif by concatenating the groups. Remove leading and trailing wildcard regions.
Ensure the first and last motif-elements have min=max. If not, delete leading and trailing terms to ensure this.
Stop

Figure 1. The ProMot algorithm.

$$Con(A,col) = \frac{freq(A,col)}{N(col)} \tag{2}$$

$$M(col) = Average(Con(A,col) \quad \forall\, A\ such\, that, Con(A,col) > 0$$

$$\sigma(col) = \sqrt{Average[\ M(col) - Con(A,col]^{\,2})} \tag{3}$$

$$Cum\_Con(A,col) = \sum Con(B,col) \quad \forall\, Con(B,col) < Con(A,col) \tag{4}$$

$$Count(A,col) = \sum 1 \quad \forall\, Con(B,col) < Con(A,col) \tag{5}$$

$$Norm\_Cum\_Con(A,col) = \frac{Cum\_Con(A,col)}{1 + Count(A,col)} \tag{6}$$

Where, *A* and *B* denote amino-acids, *Con* refers to the Contribution, *Cum_Con* refers to the cumulative sum of the Contributions, *Norm_Cum_Con* refers to the cumulative sum of the normalized contributions, *N* refers to the total number of sequences for that column (including where gaps are present), *freq(A,col)* refers to the frequency of occurrence of *A* in column *col*, *M(col)* is the mean-contribution of a column, and σ(*col*) is the Standard-Deviation of a column.

All amino-acids having their normalized cumulative contributions before a set threshold are removed from the choice-list for that group. However, as a special case, if any amino-acid has 98% or more contribution in a column, all other amino-acids are removed, and the *min* is set to 1 for that group. After this phase, the algorithm enters the

Table 3. Amino-acid classification

Classification according to the charge & polarity of the side-chain		**Classification according to the structure of the side-chain**	
Group	**Members**	**Group**	**Members**
Hydrophobic	Alanine, Glycine, Leucine, Valine, Isoleucine, Phenylalanine, Tryptophan, Methionine, Proline	Aliphatic	Glycine, Alanine, Valine, Leucine, Isoleucine
Hydrophilic	Asparagine, Glutamine, Cysteine, Serine, Threonine, Tyrosine	Hydroxyl or Sulphur/Selenium containing	Serine, Cysteine, Selenocysteine, Threonine, Methionine
Acidic	Aspartate, Glutamate	Cyclic	Proline
Basic	Arginine, Histidine, Lysine	Aromatic	Phenylalanine, Tyrosine, Tryptophan
		Basic	Arginine, Histidine, Lysine
		Acidic	Aspartate, Glutamate, Asparagine, Glutamine

final motif generation phase, which is performed using concatenation of the choice-lists of successive groups. All non-standard amino-acids are first removed from any choice-lists, and the leading and trailing wild-card regions are also removed from the generated motif. Finally, it is ensured that the starting and ending motif elements have *min=max*, by removing all elements to the left and right of the suitable starting and ending elements. The generated motif is then matched against the sample and the sample coverage is computed, and both the motif and the sample-coverage are reported to the user.

3. Materials

3.1 Sample Selection

For the purpose of motif discovery, we have worked with three protein families as summarized in Table 4. For the first test case however, the sample selected was only a subset of all the known proteins in that category. This was done to further analyze the fitness of the generated motifs to the overall population, so that we can determine whether the generated motif is flexible enough.

The SwissProt[14] database (2016_04 release) was used to find motif hits (except for HMMER, where SwissProt (2015_09 release) was used). All statistics, like sample coverage, precision, etc. were computed using SwissProt (2015_09 release), and all samples were also taken from the same release.

3.1.1 Snake Toxins

In this family, we had selected 91 short neurotoxins from the SwissProt database, from 45 species of snakes in the Elapidae family, as can be seen from Table 5. The shortest sequence was of length 58, with the longest sequence consisting of 86 amino-acids – the average length being 72. Most of these snake toxins work by binding to the

Table 4. Summary of the three test cases

#	Sample Name	No. of sequences	Min. length	Max. length	Average length	PROSITE ID (Entry Name)
1	Snake venom toxins	91	58	86	71	PS00272 (SNAKE_TOXIN)*
2	Insulin & related peptides	232	44	305	108	PS00262 (INSULIN)
3	Methylated-DNA protein-cysteine methyltransferase active site	55	108	354	173	PS00374 (MGMT)

* There are a total of 445 proteins in this category – only 20% (91 sequences) of them were selected as the sample

Table 5. List of the 91 short-neurotoxins selected from the 45 species of Elapid snakes

#	No. of proteins	Species	#	No. of proteins	Species
1	1	Acanthophis antarcticus	24	1	Micrurus pyrrhocryptus
2	4	Aipysurus laevis	25	1	Micrurus surinamensis
3	1	Austrelaps superbus	26	1	Naja annulata annulata
4	2	Bungarus fasciatus	27	4	Naja annulifera
5	3	Bungarus multicinctus	28	1	Naja christyi
6	1	Demansia vestigiata	29	2	Naja haje haje
7	1	Dendroaspis jamesoni kaimosae	30	1	Naja kaouthia
8	1	Dendroaspis polylepis polylepis	31	1	Naja melanoleuca
9	1	Dendroaspis viridis	32	2	Naja mossambica
10	2	Drysdalia coronoides	33	2	Naja nivea
11	2	Hemachatus haemachatus	34	1	Naja oxiana
12	1	Hoplocephalus stephensii	35	1	Naja pallida
13	1	Hydrophis cyanocinctus	36	1	Naja philippinensis
14	3	Hydrophis hardwickii	37	1	Naja samarensis
15	1	Hydrophis lapemoides	38	1	Notechis scutatus scutatus
16	1	Hydrophis ornatus	39	9	Ophiophagus hannah
17	2	Hydrophis peronii	40	2	Oxyuranus microlepidotus
18	1	Hydrophis schistosus	41	3	Oxyuranus scutellatus scutellatus
19	1	Hydrophis stokesii	42	1	Pseudechis australis
20	5	Laticauda colubrina	43	1	Pseudechis porphyriacus
21	3	Laticauda crockeri	44	7	Pseudonaja textilis
22	6	Laticauda laticaudata	45	2	Tropidechis carinatus
23	1	Laticauda semifasciata			

nicotinic acetylcholine receptors in the postsynaptic membrane of skeletal muscles, thus suspending muscle excitation through prevention of acetyl choline binding[15]. The PROSITE protein family identifier for this family is PS00272.

3.1.2 Insulin

In this category, we had selected 232 sequences from the SwissProt database – a group of active peptides like insulin, relaxin, insulin-like growth factors, and various other insulin-like peptides; all of which are thought to be evolutionarily related. The sequence lengths vary from 44 to 305; the average length being 109. The PROSITE protein family identifier for this family is PS00262.

3.1.3 Methylated-DNA Protein-Cysteine Methyltransferase Active Site

In this category, we had selected 55 sequences from the SwissProt database – a group of enzymes, who are responsible for the repair of DNA containing O6-alkylated guanine, which is usually formed as the mutagenic & carcinogenic effects of methylating agents[16]. After the repair process, the enzyme is irreversibly inactivated[17]. The sequence lengths vary from 108 to 354; the average length being 173. The PROSITE protein family identifier for this family is PS00374.

3.2 Alignment and Submission for Motif Discovery

The proteins were aligned using MUSCLE, and then those aligned sequences were fed into Pratt, ProMot and HMMER to obtain motifs/profiles for the respective protein families. DRIMust and MEME require unaligned sequences, and hence the raw samples were fed into those to generate the motifs/profiles. The search parameters used for Pratt and MEME are listed in Appendix A.

4. Results

4.1 Snake Toxins

In Table 6, we present the motifs generated by the various algorithms, when presented with the aligned/unaligned sequences of the short-neurotoxins (Table 3), and compare it against the PROSITE consensus motif. From the motifs, we can see that only ProMot correctly identifies 7 of 8 Cysteines involved in disulphide bonds, which are invariant in most snake venom toxins. We can also easily see that, only ProMot generated variable-length wildcards in the same motif-element, which none of the other algorithms could, and is a very much required feature as seen in the manually generated PROSITE motif. However, we see that both ProMot and Pratt fail to detect the Proline, thought to be essential for structural stability, which is present in the PROSITE consensus motif. However, we must take into consideration that the PROSITE motif is manually curated, while both ProMot and Pratt are unsupervised algorithms. We can also see that MEME cannot produce gapped motifs, while the motif generated by DRIMust is too general. With ProMot, MAX_CHOICES was set to 2, and ALLOW_GROUPING set to NO.

Table 6. Comparison between the motifs generated for snake-toxins by ProMot, Pratt, MEME & DRIMust and the PROSITE snake-toxin consensus motif (PS00272). HMMER is not included in the table as it does not generate motifs but instead builds a HMM profile

	ProMot	**Pratt (v2.1)**	**PROSITE PS00272**	**MEME (v4.11)**	**DRI Must**
Motif	C-x(4,17)-C-x(4,6)-C-x(9,21)-G-C-x(1,3)-C-x(3,11)-C(2)-x(5)-N	C-C-x(2)-[DEN]-x-[CS]-N	G-C-x(1,3)-C-P-x(8,10)-C-C-x(2)-[PDEN]	W-R-D-H-R-G-T-I-I-E-R-G-C	G-C-G-C
No. of motif components	14	7	10	13	4
Motif length (with dashes)	61	23	38	25	7
Motif length (w/o dashes)	48	17	29	13	4
Min. matchable region	35	8	18	13	4
Max. matchable region	72	8	22	13	4

4.2 Insulin

In Table 7, we present the motifs generated by the various algorithms, when presented with the aligned/unaligned sequences of the insulin family, and compare it against the PROSITE consensus motif. We see that Pratt and DRIMust failed to generate any motifs at all – the latter due to the fact that the insulin proteins contain two non-standard amino acids B & Z. The motif generated by MEME is too long (and general) to be submitted to ScanProsite[18] for searching against SwissProt, and hence the count matrix, rather than the motif had to be submitted to FIMO[19]for searching in SwissProt. Comparing the ProMot and PROSITE consensus motifs, we see that ProMot correctly identified four of the Cysteines involved in disulphide bonds, which are also present in the PROSITE motif. With ProMot, MAX_CHOICES was set to 8, and ALLOW_GROUPING set to YES.

4.3 Methylated-DNA Protein-Cysteine Methyltransferase Active Site

In Table 8, we present the motifs generated by the various algorithms, when presented with the aligned/unaligned sequences of the MGMT family (methylated-DNA protein-cysteine methyltransferase active site), and compare it against the PROSITE consensus motif. We see that yet again, DRIMust failed to generate any motifs at all. MEME generated a count-matrix (hence the motif is not listed in the table), which was fed into FIMO for searching. With ProMot, MAX_CHOICES was set to 3, and ALLOW_GROUPING set to NO.

Table 7. Comparison between the motifs generated for insulin by ProMot, Pratt, MEME & DRIMust and the PROSITE insulin-family consensus motif (PS00262). HMMER is not included in the table as it does not generate motifs but instead builds a HMM profile

	ProMot	Pratt (v2.1)	PROSITE PS00262	MEME (v4.11)	DRIMust
Motif	[CDEHRST]-[EKQSY]-F(0,1)-C-C-{CPW}-{ALFP}-[AG](0,1)-x-C-[DENST]-x(3,4)-[AEKLQRS]-x(0,1)-F(0,3)-[FL](0,1)-[ALY]-[ILM](0,1)-[CX]	Failed to generate any motifs	C-C-{P}-{P}-x-C-[STDNEKPI]-x(3)-[LIVMFS]-x(3)-C	Motif generated as a probability matrix	Failed due to the presence of non-standard amino-acid codes: B & Z
No. of motif components	19	-	11	17	-
Motif length (with dashes)	126	-	47	281	-
Motif length (w/o dashes)	108	-	37	265	-
Min. matchable region	15	-	15	18	-
Max. matchable region	24	-	15	18	-

Table 8. Comparison between the motifs generated for methylated-DNA—protein-cysteine—methyltransferase-active-site by ProMot, Pratt, and the PROSITE MGMT-family consensus motif (PS00374). HMMER is not included in the table as it does not generate motifs but instead builds a HMM profile, DRIMust failed to generate any motifs, and MEME generated a count-matrix

	ProMot	Pratt (v2.1)	PROSITE PS00374
Motif	G-x(4)-Y-x(3)-[AV]-x(5)-[KP](0,1)-x(5,7)-[AG]-x(5)-[LN]-x(6)-[AP]-[CW]-H-R-[IV]-x(15,23)-[KQ]-x(3)-L-x(2)-E	H-R-[IV]-[ILV]	[LIVMF]-P-C-H-R-[LIVMF](2)
No. of motif components	23	4	6
Motif length (with dashes)	107	14	26
Motif length (w/o dashes)	85	11	21
Min. matchable region	61	4	7
Max. matchable region	72	4	7

5. Discussion

5.1 Terminology

The motif matching statistics have been presented in Table 9, 10 and 11. For analyzing these, it is crucial to be aware of some terminology used there:

5.1.1 Total Hits (or Number of Sequences Matched)

It is the total number of sequences that match the generated motif, when that motif is used for searching against the SwissProt database.

Table 9. Snake toxin hit statistics by the motifs of Table-6 and HMMER profile, on the SwissProt database

Category/Motif	ProMot	Pratt (v2.1)	PROSITE PS00272	HMMER (v3.0)	MEME (v4.11)	DRI Must
No. of sequences matched	419	549	398	502	18	561
Short neurotoxins	95	96	79	97	17	72
Long neurotoxins	45	51	53	59	0	0
Elapitoxins	25	23	25	27	0	0
Alpha neurotoxins	7	7	7	7	0	6
Weak neurotoxins	18	13	13	14	0	1
Weak toxins	15	16	13	15	0	2
Cobrotoxins	7	7	7	7	0	7
Cytotoxins	87	93	92	92	0	0
Cardiotoxins	8	5	5	10	0	0
Three-finger toxins	18	15	11	26	1	10
Erabutoxins	3	3	3	3	0	3
Hemachatoxins	0	1	1	0	0	0
Bungarotoxins	13	14	13	14	0	0
Pseudonajatoxins	2	2	2	2	0	0
Other toxins & venom-like proteins	64	66	67	84	0	17
Other proteins (false positives)	12	137	7	45	0	443
True Positives	407	412	391	457	18	118
Sample coverage	100%	100%	100%	100%	17.6%	74.7%
Precision = (True positives / Total hits)	97.1%	75.0%	98.2%	91.0%	100%	21.0%

Table 10. Insulin hit statistics by the motifs of Table-7 and HMMER profile, on the SwissProt database

Category/Motif	ProMot	PROSITE PS00262	HMMER (v3.0)	MEME (v4.11)
No. of sequences matched	199	230	241	21,417
True Positives	183	222	232	230
False Positives	16	8	9	21,187
False Negatives	49	10	0	2
Sample coverage	78.9%	95.7%	100%	99.1%
Precision = (True positives / Total hits)	92%	96.5%	96.3%	1.1%
Recall = (True positives / (True positives + False negatives)	78.9%	95.7%	100%	99.1%

Table 11. MGMT hit statistics by the motifs of Table-8 and HMMER profile, on the SwissProt database

Category/Motif	ProMot	Pratt (v2.1)	PROSITE PS00262	HMMER (v3.0)	MEME (v4.11)
No. of sequences matched	57	7,895	67	211	23,581
True Positives	55	55	53	55	55
False Positives	2	7,840	14	156	23,526
False Negatives	0	0	2	0	0
Sample coverage	100%	100%	96.4%	100%	100%
Precision = (True positives / Total hits)	96.5%	0.7%	79.1%	26.1%	0.2%
Recall = (True positives / (True positives + False negatives)	100%	100%	96.4%	100%	100%

5.1.2 Sample and Sample Coverage

Usually all known proteins of a family are used for generating the motif (Test cases 2 and 3), but in some cases (Test case 1) it is imperative to use only a subset because it better mimics the real life scenario when we do not yet know all proteins which belong to a family, and we must generate a motif and use it to search for other proteins similar to the ones we have chosen. The sample therefore may either be a subset or be the complete set of sequences generally believed to constitute a given protein family. Sample coverage is then the percentage of proteins matched by the motif (generated from that sample) in that same sample.

5.1.3 True Positives

It is the number of proteins matched by the motif that are believed to be in a given protein family.

5.1.4 False Positives

It is the number of proteins matched by the motif that are not believed to be in a given protein family.

5.1.4 False Negatives

It is the number of proteins believed to be in a given protein family, but are not matched by the generated motif. In other words, it is the difference between the number of proteins believed to be in a protein family and the number of True Positives.

It must be noted that the actual number of proteins in any given protein family is always unknown. However, for all possible purposes of the various statistics presented in the tables below, only those proteins are taken into consideration, which are widely believed to be in the said protein family. The notion of 'wide belief' may appear to be vague, and as such, the criteria used on PROSITE for determining the true and false positives, and false negatives, for each of the 3 protein families tested in this paper, have been applied here as well.

5.2 Characteristics of the Generated Motifs

5.2.1 Snake Toxins

All the motifs of Table 6 were run against the SwissProt database using the ScanProsite tool, and the match statistics are presented in Table 9. From the table, we can see that ProMot is much more accurate than Pratt, and almost as accurate as the PROSITE consensus motif. However, the primary advantage of Pratt over ProMot is that it can generate motifs from unaligned protein sequences as well, while ProMot requires an aligned set of sequences. From the table, we can see that ProMot outperformed all other algorithms but fell short of matching the PROSITE motif marginally in terms of precision. MEME had a too low sample-coverage to be useful, while the precision of DRIMust was extremely low.

5.2.2 Insulin

All the motifs of Table 7 were run against the SwissProt database and the match statistics are presented in Table 10. The motif produced by MEME was too long to be submitted to ScanProsite, and hence the count-matrix generated by the program was submitted to FIMO for the search. Pratt and DRIMust are excluded from the table because both of these failed to generate any motifs. As far as sample coverage (which is the percentage of matches against the sequences in SwissProt *known* to be in the Insulin family), ProMot outperforms even the PROSITE motif. However, it also reports too many false positives. The HMMER algorithm clearly outperforms all the others, both in terms of sample coverage and almost in terms of precision. The only disadvantage of the HMMER tool is the lack of any motif returned to the user, thus prohibiting any visual inference of the general protein family structure directly from the profile. The results produced by MEME in this category can be ignored as it reports too many false positives to be useful, though its sample coverage is impressive.

5.2.3 Methylated-DNA Protein-Cysteine Methyltransferase Active Site

All the motifs of Table 8 were run against the SwissProt database and the match statistics are presented in Table 11. The motif produced by MEME was too long to be submitted to ScanProsite, and hence the count-matrix generated by the program was submitted to FIMO for the search. Pratt and DRIMust are excluded from the table because both of these failed to generate any motifs. From the table we can see that our proposed algorithm outperforms all other algorithms, even the PROSITE motif itself, both in terms of sample-coverage as well as precision.

5.3 Runtime Performance of the Algorithm

The running times of the various algorithms are tabulated in Table 12. The time complexity of the ProMot algorithm is $O(n.m+m^2)$, where n is the total number of sequences, and m is the length of each sequence after alignment. Its auxiliary space complexity is $O(n)$. As we can see from the table, our algorithm is easily the fastest algorithm among all other algorithms except Pratt, with which we cannot compare as its running-time precision was not available down to microsecond level. We must also keep in mind that Pratt, DRIMust and the MEME algorithms were run on servers with better specifications, while HMMER and ProMot were run locally on an Intel Celeron 1.6GHz processor with 2GB memory.

Table 12. Running times of the various algorithms (in seconds)

Algorithm	Snake Toxins	Insulin	MGMT
HMMER	0.60	2.60	2.42
MEME (on server)	5.79	71.66	5.93
Pratt (on server)	0	FAILED	0
ProMot	0.03	0.23	0.06
DRIMust (on server)	Not available	FAILED	FAILED

5.4 Comparison of the Test Case Results

A visual comparison of the algorithm performances can be seen in Figure 2–4, for the three test-cases showing their sample-coverage and precision. Also, as we can see from the three test-cases presented, ProMot also never fails to generate motifs.

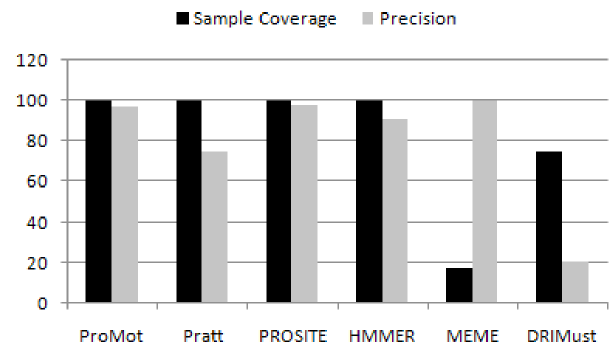

Figure 2. Comparison of sample-coverage and precision of the various algorithms on the snake toxins family.

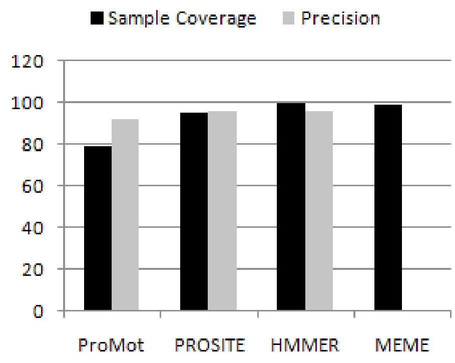

Figure 3. Comparison of sample-coverage and precision of the various algorithms on the insulin family.

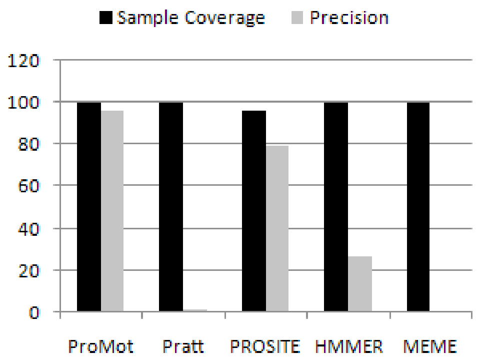

Figure 4. Comparison of sample-coverage and precision of the various algorithms on the MGMT family.

5.5 Software Availability

The algorithm (version 4.0) has been implemented in Java 8, and the tool requires Java to execute. The software has been made freely available on sourceforge. Java is also freely available for download from the Oracle website. To use the software, one has to download the ProMot JAR (Java Archive) from sourceforge, after installing Java, and execute the software providing a set of aligned sequences as input. The input file must be in FASTA format. For ease of use, a sample aligned FASTA dataset has also been included for download. When run without any parameters, the software shall display the proper usage with the list of parameters to specify. The output can be seen on the console, or can be redirected to an output file.

6. Conclusion

Motif discovery from a set of protein sequences has several advantages in bioinformatics. Firstly, it helps us identify the general structure of all the proteins in that family by identifying the invariant residues. Secondly, it helps to predict the function of that protein, and this has several appications[20-25]. Thirdly, it is much easier and memory-efficient to store a single motif, than store a large set of sequences to analyze later, when we can dynamically generate the larger set by matching that motif against any database.

The most widely used motif discovery algorithms are HMMER, MEME and the Pratt algorithms; however, as we have seen, our proposed method generates better and more accurate motifs than Pratt, and MEME, while it outperforms HMMER in two of three occasions in terms of precision. ProMot is extremely fast, and hence, the user may first generate an alignment using any available MSA algorithm, and then feed those sequences to ProMot to get the desired motif. A huge advantage of ProMot is that the user-parameters are much more intuitive and easy for the user to specify and does not require prior knowledge about the probable motif. We hope that ProMot shall greatly advance the state-of-the-art in motif discovery for protein families.

7. References

1. Bairoch A. PROSITE: a dictionary of sites and patterns in proteins. Nucleic Acids Research. 1991; 19:2241–5. Crossref, PMid:2041810 PMCid:PMC331358
2. Ogiwara A, Uchiyama I, Yasuhiko S, Kanehisa M. Construction of a dictionary of sequence motifs that characterize groups of related proteins. Protein Eng. 1992; 5:479–88. Crossref, PMid:1438158
3. Saqi MAS, Sternberg MJE. Identification of sequence motifs from a set of proteins with related function. Protein Engineering. 1994; 7:165–71. Crossref
4. Wang JTL, Marr TG, Shasha D, Shapiro BA, Chirn GW. Discovering active motifs in sets of related protein sequences and using them for classification. Nucleic Acids Res. 1994; 22(14):2769–75. Crossref, PMid:8052532 PMCid:PMC308246
5. Koza JR, Andre D. Automatic discovery of protein motifs using genetic programming. 1996. p. 542.
6. Smith RF, Smith TF. Automatic generation of primary sequence patterns from sets of related protein sequences. Proc Nutl Acud Sei. 1990; 87:118–22. Crossref
7. Henikoff S, Henikoff JG. Automatic assembly of protein blocks for database searching. Nucleic Acids Res. 1991; 9:6565–72. Crossref
8. Jonassen I, John FC, Desmond G. Higgins. Finding flexible patterns in unaligned protein sequences. Protein science. 1995; 4(8):1587–95. Crossref, PMid:8520485 PMCid:PMC2143188
9. Durbin R. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press; 1998. p. 1–366. Crossref
10. Bailey TL. MEME: discovering and analyzing DNA and protein sequence motifs. Nucleic acids research. 2006; 34(2):369–73. Crossref, PMid:16845028 PMCid:PMC1538909
11. Leibovich L, Yakhini Z. Efficient motif search in ranked lists and applications to variable gap motifs. Nucleic acids research. 2012; 40(13):5832–47. Crossref, PMid:22416066 PMCid:PMC3401424
12. Leibovich L. DRIMust: a web server for discovering rank imbalanced motifs using suffix trees. Nucleic acids research. 2013 Jul; 41:174–9. Crossref, PMid:23685432 PMCid:PMC3692051
13. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research. 2004 Mar; 32(5):1792–7. Crossref, PMid:15034147 PMCid:PMC390337
14. Bairoch A, Boeckmann B. The SWISS-PROT protein sequence data bank. Nucleic Acids Res. 1997; 25(1):31–6. Crossref, PMid:9016499 PMCid:PMC146382
15. Hodgson WC, Janith CW. In vitro neuromuscular activity of snake venoms. Clinical and Experimental Pharmacology and Physiology. 2002; 29(9):807–14. Crossref
16. Lindahl T. Regulation and expression of the adaptive response to alkylating agents. Annual review of biochemistry. 1988; 57(1):33–157. Crossref, PMid:3052269
17. Samson L. The suicidal DNA repalr methyltransferases of microbes. Molecular microbiology. 1992 Apr; 6(7):825–31. Crossref, PMid:1602962
18. De CE. ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. Nucleic acids research. 2006 Jul; 34(2):362–5.
19. Grant CE, Timothy LB, Noble WF. FIMO: scanning for occurrences of a given motif. Bioinformatics. 2011 Apr; 27(7):1017–8. Crossref, PMid:21330290 PMCid:PMC3065696
20. Priya EM. Adverse Effects of Combinatorial Therapy of Vildagliptin and Insulin on Cardiac Tissues in Diabetic Rats. Indian Journal of Science and Technology. 2016 Jan; 9(3):1–5. Crossref
21. Suresh MP, Juliet AV. Design of a Valveless Micro-Pump for Variable Rate of Insulin Delivery. Indian Journal of Science and Technology. 2016 Nov; 9(44):1–5. Crossref
22. Sultana RR, Zafarullah SN, Kirubamani NH. Insulin Response of Diabetic Pregnant Women: Analysis of Saliva by FTIR Study. Indian Journal of Science and Technology. 2012 Mar; 5(3):1–7.

23. Son HK, Choi HS, Park JR. Modulation of Oral Epithelial Cell Properties by Insulin. Indian Journal of Science and Technology. 2015 Jan; 8(S1):1–5. Crossref

24. Kusakabe M, Takei Y, Luckenbach JA. Relaxin-3 and Relaxin/Insulin-like Family Peptide Receptor 3 in Rainbow Trout: Sites of Gene Expression and Changes in Messenger RNA Levels during Spermatogenesis in Testes. Indian Journal of Science and Technology. 2011 Aug; 4(S8):1–2.

25. Jeyabalan S, Raj VC. A Novel Technique for Analysis of Protein to Protein Interaction using Efficient Minimum Spanning Tree Techniques. Indian Journal of Science and Technology. 2016; 9(41):1–5. Crossref

Appendix A: Search Parameters

The default parameters were used for Pratt when generating motifs as listed below:

PATTERN CONSERVATION:

C%: min Percentage Seqs to Match	100.0

PATTERN RESTRICTIONS :

PP: pos in seq [off,complete,start]	off
PL: max Pattern Length	50
PN: max Nr of Pattern Symbols	50
PX: max Nr of consecutive x's	5
FN: max Nr of flexible spacers	2
FL: max Flexibility	2
FP: max Flex.Product	10
BI: Input Pattern Symbol File	off
BN: Nr of Pattern Symbols Initial Search	20

PATTERN SCORING:

S: Scoring [info,mdl,tree,dist,ppv]	info

SEARCH PARAMETERS:

G: Pattern Graph from [seq,al,query]	al
E: Search Greediness	3
R: Pattern Refinement	on
RG: Generalise ambiguous symbols	off

Default parameters for MEME were used as well, for generating the motifs, and are listed below:

model: mod=oops, nmotifs=1, evt=inf, object function= E-value of product of p-values
width: minw=6, maxw=58
width: wg=11, ws=1, endgaps=yes
theta: spmap=pam, spfuzz=120
global: substring=yes, branching=no, wbranch=no
em: prior=dmix, b=0, maxiter=50, distance=1e-05

For all other algorithms as well, default parameters were used. Parameters used for ProMot are stated in the text against each motif generated.