A Method to Find Sequentially Separated Motifs in Biological Sequences (SSMBS)

Chetan Kumar[1], Nishith Kumar[1], Sarani Rangarajan[1], Narayanaswamy Balakrishnan[2], and Kanagaraj Sekar[1,2,⋆]

[1] Bioinformatics Centre (Centre of excellence in Structural Biology and Bio-computing)
Tel.: +91-80-22933059/22932469/23601409; Fax: +91-80-23600683/23600551
sekar@physics.iisc.ernet.in, chetan-k@northwestern.edu
nishith_iitd@yahoo.co.in, sarani.rangarajan@gmail.com
[2] Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560 012, India
{balki,sekar}@serc.iisc.ernet.in
http://www.physics.iisc.ernet.in/~dichome/sekhome/index.html

Abstract. Sequence motifs occurring in a particular order in proteins or DNA have been proved to be of biological interest. In this paper, a new method to locate the occurrences of up to five user-defined motifs in a specified order in large proteins and in nucleotide sequence databases is proposed. It has been designed using the concept of quantifiers in regular expressions and linked lists for data storage. The application of this method includes the extraction of relevant consensus regions from biological sequences. This might be useful in clustering of protein families as well as to study the correlation between positions of motifs and their functional sites in DNA sequences.

Keywords: Regular expressions, protein and nucleotide sequences, sequence motifs.

1 Introduction

Research on proteins and DNA has revealed that specific motifs in biological sequences exhibit important characteristics [1]. This has spurred the development of computational methods to search for sequence motifs of biological significance. Further, the exponential rise in the volume of protein and nucleotide sequences has necessitated the development of algorithms that are both time and space efficient to make optimum use of available computational resources. Here, an efficient method is proposed that locates all occurrences of motifs of biological interest in a specific order using the concept of quantifiers in regular expressions.

⋆ Corresponding author.

M. Chetty, A. Ngom, and S. Ahmad (Eds.): PRIB 2008, LNBI 5265, pp. 13–27, 2008.

We refer to motifs that occur in a particular order as "sequentially separated motifs", since they could be separated by intermediate amino acid residues or nucleotides.

Recent studies have considered sequentially separated motifs as a method for classifying DNA sequences based on the presence and relative positions of a few transcription factor (TF) binding sites. These binding sites are of such importance that several algorithms and online tools are available for their detection[2],[3],[4],[5] . Binding sites are relatively short stretches of DNA, normally 5 to 35 nucleotides long and occur as consensus regions or well conserved regions called motifs. It has been established in literature that binding sites are often found in a well-ordered and regularly spaced manner [6],[7],[8] . In prokaryotic organisms, the binding sites are located predominantly in the region that extends about 300 to 600 nucleotides upstream of the transcription start site, in the promoter regions. However, in eukaryotic organisms, the binding sites, called cis-regulatory modules (CRMs), usually occur in a fixed arrangement and are distributed over very large distances. A detailed explanation of eukaryotic promoters can be found in literature [9],[6]. A eukaryotic promoter is considered to comprise of three CRMs, each having one or more TF binding sites. Since each CRM has a different function, it will be helpful to have a method that can locate the distribution of the occurrences of the three CRMs in the order in which they exist in the sequence. Further, repeated occurrences of CRMs in the DNA sequence might lead to alternate modes of binding by the same protein, thereby regulating transcriptional activity. In addition, it may lead distinct proteins to recognize the identical CRMs occurring at different positions in the sequence. Also, if the signature motifs for trans-regulatory modules are known, they too can be detected to achieve a more complete understanding of the the structure of the gene and its regulation.

Furthermore, sequentially separated conserved motifs have been used to categorize new and unknown protein structures. For instance, the classification of T6PP as a member of the haloacid dehalogenase (HAD) superfamily is based on the presence of three highly conserved motifs that are found in all enzymes belonging to the HAD family. The three motifs are DXXX(V/T), followed by (S/T)GX, and finally $K(X)_{(16-30)}$(G/S)(D/S)XXX(D/N), where X denotes a wild card symbol that can be substituted by any of the 20 amino acids and G/S signifies the presence of G or S at the particular position in the motif [10],[11]. The HAD superfamily is further subdivided into three structural groups based on the length of the sequence between the motifs [12]. Thus, it can be concluded that in proteins, the intermediate sequences that separate the sequential motifs are also biologically significant. The concept of sequentially separated motifs finds an important application in remote homology detection of proteins. Homology is generally established by sequence similarity. In the past two decades, many methods for measuring sequence similarity have been developed. The two most popular methods are the Smith-Waterman algorithm [11] and its faster counterpart, BLAST[13]. Protein sequence motifs can offer an alternative way of detecting sequence similarity. By closely studying highly conserved sequence

motifs, important clues to a protein function might be revealed even if it is not globally similar to any known protein [14]. In addition, the sequentially separated motifs for most catalytic sites and binding sites are conserved over much wider taxonomic distances and evolutionary time than the protein sequences themselves [15]. Thus, it can be deduced that motifs that are found to occur in a particular order could represent functionally important regions such as catalytic sites, binding sites, protein-protein interaction sites and structural motifs.

In view of the biological relevance of sequentially separated motifs, a need is felt to develop a method that can detect the occurrences of the motifs in large sequence databases efficiently. The performance of such a method designed to solve this problem should be judged according to the following criteria:

1. Efficiency: To analyze large nucleotide and proteins sequences (e.g. the human chromosome 1 contains 240 million nucleotide bases and the proteome of *A. thaliana* more than 7,000 protein sequences), the space and time complexity of the method must scale linearly with the sequence length and the number of sequences. Further, the method should also minimize the number of iterations and comparisons required to report all occurrences of the motifs.
2. Flexibility: The motifs should be specified using regular expressions.
3. Accuracy: To identify all locations, including degenerate occurrences and overlapping occurrences.
4. User-Friendliness: It should be simple to use, platform independent and display results in an elegant and easily comprehensible manner.

1.1 Existing Algorithms

Two types of pattern matching algorithms are commonly used in biology:

scan_for_matches. [16] brought on a series of other software and algorithms, including PatScan [17] which searches a dataset for matches against a query pattern. PatSearch [18] has added features such as the assessment of the statistical significance of pattern hits using a Markov chain simulation. The results of these programs display the entire substring that contains the motif provided by the user but do not explicitly indicate the individual occurrences of the motifs. Due to this, the user needs to manually delineate the intermediate residues that separate the motifs.

grep-based programs. An example of which is eMOTIF-SCAN, a program which uses the agrep tool that supports matching and regular expression. However, it searches only against the eMOTIF database of protein sequence motifs [19].

The program, Scansite 2.0 [20] searches for up to two motifs and looks for the occurrences of these motifs in no particular order of arrangement. Motif Scan [21] searches for motifs against protein profile databases including Prosite [1] and Pfam [22], and, thus does not provide the users with the option to enter their own motifs. Though most of the above mentioned programs work efficiently with

protein sequences, they do not perform well with large sequences. In most cases, the programs do not execute to completion for very large nucleotide sequences (150 million bp).

Furthermore, the pattern search present in the PIR database is also extremely efficient for a single motif (or when a number of motifs can be combined to a single motif). However, when the specific number of residues between two or more motifs cannot be identified, two separate PIR pattern searches must be run and the results compared either manually or through a program written specifically to obtain the required sequences from the output of the two searches. This process becomes much more complicated when more than two motifs are being searched in order in a set of sequences. Finally, in SSMBS, the database of sequences to be searched for the motif can be specified or uploaded. On the other hand, in the PIR pattern search, only two options for databases exist if a search for a user-defined motif must be carried out: UniPotKB and UniRef100. Thus, when the user wishes to find a number of motifs in order (with an unknown of large number of residues separating the motifs) in a user specified database, SSMBS is the only available option.

2 Materials and Methods

2.1 Basic Definitions

If S = {A, C, G, T, U} is the alphabet defined for nucleotide sequences (U for RNA) and S = {A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V} is the alphabet defined for amino acid sequences, then let S, defined over S, represents the sequence in which the sequentially separated motifs are to be located. Further, let n = |S| i.e. length of S. Let m be the number of input sequences.

S[i] denotes the i^{th} character of S, for i ∈ [1,n]. For i ≤ j ≤n, S[i,j] denotes the substring of S starting with the i^{th} and ending with the j^{th} character of S. Thus, the length of S[i,j] is j - i + 1.

Let M = { $motif_i | 1 \leq i \leq 5$ } be the set of up to five motifs entered by the user and k = |M|, i.e. number of motifs, where M is defined over S.

L denotes a linked list whose elements comprise of many other linked lists, each called L'. L' contains the starting positions of the occurrences of the M_i such that L' = { $(s_i),(s_{i+1}),\ldots,(s_k)$ | s_i is starting position of M_i; $1 \leq s_i \leq n$; $1 \leq i \leq 5$ }.

2.2 Use of Quantifiers

Quantifiers, as the name implies, express quantity i.e. how much or how many. They are used in pattern matching since they allow us to control the amount of text in a sequence that is to be matched against a pattern. Quantifiers have already been implemented in several programming languages including JAVA and Perl and they are an integral part of regular expression matching. In biological

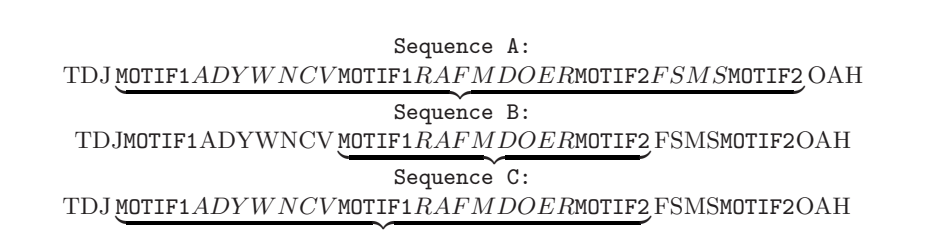

Fig. 1. Sequences A, B and C represent the three different query patterns [(.*), (.?) and (.*?) respectively] used to simultaneously locate the two motifs, motif1 (solid block) and motif2 (grey box) in that order

$$\underbrace{[MOTIF1](.*?)[MOTIF2]...[MOTIF_K]}_{EXPR}$$

Fig. 2. $EXPR$ represents the combined query pattern which is formed by appending the $(.*?)$ quantifier between the k motifs entered by the user

sequences, they can be used to match complex motifs that are defined using regular expressions.

'*' is a greedy quantifier which tries to match as much text as possible in the query string. However, '?' is a reluctant quantifier which tries to match as less text as possible. In this method, 'minimal matching' is utilized: the two quantifiers, '*' and '?' are coupled in the order (.*?) and appended between the two motifs motif1, motif2 $\in$ M , such that: [motif1](.*?)[motif2]. This enables the detection of both occurrences simultaneously (Figure 1c).

This concept can be further extended to simultaneously detect the first occurrences of any number of motifs. This can be achieved by appending '(.*?)' between the motifs to form an expression EXPR as shown in Figure 2.

EXPR suggests that the SSMBS (Sequentially Separated Motifs in Biological Sequences) method appends the (.*?) quantifier after every motif till the k^{th} motif. At the time of execution, the user is asked to specify whether the sequence file provided contains amino acids or nucleotides. The method exploits the technique explained above to search for motifs in a defined order in proteins sequences. However, in case of large nucleotide sequences (>100,000 bp), the method follows the divide and conquer approach, as outlined in the subsequent sections.

2.3 Amino Acid Sequences

Let us consider a case in which the user enters five sequentially separated motifs and a set of 10,000 amino acid sequences. As SSMBS reads each sequence, it first checks whether there exists, in that sequence, at least a single occurrence of the five motifs in the order specified by the user. If a match is found, then it attempts to find all occurrences of the motifs in that particular sequence. If there does not exist any match, it moves to the next sequence and performs the

same check. To find all occurrences of the five sequentially separated motifs in these sequences, SSMBS first simultaneously locates all occurrences of the last two motifs i.e. motif4 and motif5, followed by occurrences of motif3, motif2 and finally motif1. An explanation of this procedure for k motifs follows.

Locating all occurrences of k motifs. For k motifs entered by the user (k $\leq$ 5), the last two motifs are $\text{motif}_{(k-1)}$ and $\text{motif}_k \in$ M respectively. Let 'R' be the set of remaining motifs i.e. motif_1 to $\text{motif}_{k-2} \in$ M. The terms 'last two motifs' and 'R' hold significance as they divide the method into two fundamental parts: first, finding all occurrences of the last two motifs and second, finding all occurrences of the R motifs in desired order. To locate all occurrences of the last two motifs, the method appends the '(.*?)' quantifier between the motifs to locate their occurrences simultaneously in the order, (k-1)$^{\text{th}}$ motif followed by the k$^{\text{th}}$ motif. The matching performed by the method returns the starting index of $\text{motif}_{(k-1)}$ and the end index of $\text{motif}_{(k)}$. Further, a series of iterations are performed to extract all occurrences of the two motifs in the specified order. The procedure of the first step is illustrated in the form of a pseudo code as shown below:

```
 Pat = [motif_(k−1)](.*?)[motif_k], SEQ = S
do {
if(find(Pat)) // returns true if a match is found.
{ start-index-m_(k−1) = start(); // returns starting index of m_(k−1).
end-index-m_k = end(); // returns ending index of m_k.
start-index-m_k = start(matcher(motif_k,SEQ.substring(start(),end());
// searches for motif_k only at the end of the substring.
SEQ.substring(start(),end());
Linked list changes ();
SEQ = S.substring(start-index-m_k−1, start-index-m_k-1)
+ S.substring(start-index-m_k+1,n);
```

$$\underbrace{\underbrace{MOTIF_{K-1}NDRKEMOTIF_{K-1}LVAY}\underbrace{MOTIF_kAMTEMOTIF_kLGL}}_{SEQ}$$

```
}
else { SEQ = S.substring(start-index-m_k−1 + 1, n);
```

$$\underbrace{MOTIF_{k-1}NDRKEMOTIF_{k-1}LVAYMOTIF_KAMTELPMOTIF_KLGL}$$

```
}
} while(no more matches of Pat can be found)
}
```

The second step begins when no more occurrences of the last two motifs can be found. In this step, all occurrences of the k motifs are found in the following order: (k-p)$^{\text{th}}$ motif (where p = 2,3,. . . ,(k-2)) to 1$^{\text{st}}$ motif. Thus, while searching for the occurrences of the (k-p)th motif, the method has already obtained all the occurances of the (k-p+1)th to kth motifs. All occurrences of (k-p+1)$^{\text{th}}$ to k$^{\text{th}}$ motifs are stored as a linked list L' in the form (start positions of (k-p+1)$^{\text{th}}$, (k-p+2)$^{\text{th}}$,. . . . . , k$^{\text{th}}$ motifs). To update these ordered sets by appending the

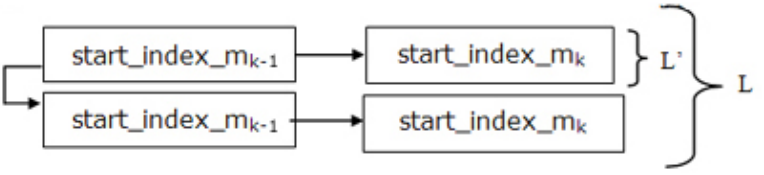

Fig. 3. Linked List appending and changes

start position of the (k-p)th motif at the beginning of L', it goes on comparing the end index of the position of the motif being dealt with, which is the (k-p)thmotif, with the first entry of every ordered set. The comparison is made at every iteration in which a new occurrence of motif$_{(k-p)}$ is detected. After successfully attaching the start index of motif$_{(k-p)}$ to L', the method appends L' to L.

For the subsequent iterations of this step, SSMBS retains only those elements or ordered sets to which the append was carried out successfully. Thus, at every iteration, the unwanted sets are eliminated, thereby shortening the size of the linked list L, to be searched in the iterations to follow (Figure 3).

2.4 Nucleotide Sequences

Unlike amino acids sequences, nucleotide sequences are very large often comprising of millions of bases. Their large size poses a major challenge in locating sequentially separated motifs because it is a memory exhaustive process. Accordingly, SSMBS adopts a divide and conquer strategy, breaking down the large sequence into small fragments comprising of 3,500 nucleotides. The value of 3,500 nucleotides per fragment is an optimal value that was heuristically determined after considering the time taken by the program implementing this method for varying sizes of the fragments. Let the fragments be denoted by F_s where s ranges from 1 to (n/3500 + 1). The method begins locating the occurrences of the sequentially separated motifs by traversing each fragment starting from F_1. The fragment in which the first occurrence of motif$_1$ is detected is marked F_m. Attempts to detect the occurrences of other motifs are carried out only in the fragments that follow F_m. In addition, the method also checks for any occurrences of the motifs that might overlap between regions common to two consecutive fragments, say F_a and F_{a+1} where $a < (n/3500 + 1)$. It does so by searching for an occurrence of either of the k motifs in the string $F_a + F_{a+1}$('+' denotes concatenation) and confirming whether the starting position of the substring that matches any of the motifs is less than the length of F_a i.e. $|F_a|$ and the ending position is greater than $|F_a|$. Finally, the method collates all occurrences of the k motifs and displays the results by traversing the linked list L, which stores the individual occurrences of the k motifs as in the case of amino acids sequences explained earlier.

Locating overlapping occurrences of motifs. The proposed method locates all overlapping occurrences of a motif as well and thus misses no occurrence. For instance, the motif M_{exmpl} = ATA{3,5} can be found to occur six times in a sequence of the form ATAATAATAATAATA i.e three occurrences of ATAATAATA, two occurrences of ATAATAATAATA and single occurrence of ATAATAATAATAATA. To report such overlapping occurrences, the method initially attempts a greedy match to find an occurrence of the M_{exmpl} in the sequence. For every occurrence of M_{exmpl}, SSMBS then attempts a reluctant match to find occurrences of the motif that might exist within or that overlap with the matched string that was returned as a result of the greedy search. This is achieved by appending the reluctant quantifier '?' to M_{exmpl} to form the new expression M_{exmpl}' = ATA{3,5}? Now, SSMBS matches M_{exmpl}' against the substring that matches M_{exmpl}. Thus, the reluctant match returns (ATAATAATA: 1 to 9) as the first overlapping occurrence. Successive iterations of this step return all possible overlapping occurrences.

2.5 Time Complexity

The computational complexity of SSMBS method is explained based on the following points:

1. **Complexity with regard to number of proteins sequences:** The SSMBS method searches for occurrences of k motifs only in those sequences that have at least one occurrence of EXPR. As explained earlier, EXPR detects the ordered occurrence of k motifs in O(n) time, where n is the length of the sequence. If there are m sequences in all, then in O(mn) time, the method searches for all sequences that have at least one occurrence of EXPR. Hence, the method scales linearly with the number of input sequences. This is notable especially in the context of the exponential rise in the size of sequence databases.
2. **Complexity with regard to locating all occurrences in a given sequence:** The method is able to detect all ordered occurrences of k motifs in k-1 scans of the sequence, as compared to k scans in a brute force approach. Further, as the computation grows, it optimizes by reducing the length of the query sequence based on motif positions located in previous iterations. For instance, while searching for the $motif_R$, the algorithm searches only till the last occurrence of $motif_{R+1}$ in the sequence. Specifically, the performance of the method is bounded polynomially by $O(n^{k-1})$.
3. **Complexity specifically for nucleotide sequences:** By following the divide and conquer strategy in nucleotide sequences, the method successfully avoids the out of memory problem no matter how large the nucleotide sequence is. As in the case of proteins sequences, the complexity of the algorithm scales linearly with the size of nucleotide sequence. Thus, the algorithm can be applied to search for specific regions in entire genome of different organisms.

3 Biological Applications

3.1 Motifs Specified Using Regular Expressions

Motifs with biological importance often occur with some mutations or substituted residues in the sequence. Thus, regular expressions are used to specify such motifs in SSMBS. This process is quite similar to that found in the PIR pattern search. However, one main difference between the SSMBS algorithm and the PIR pattern search is that SSMBS can search for multiple motifs in a particular order, while PIR's pattern search is limited to those patterns in which the number of intervening residues between two motifs is at least approximately known. A few examples are:

1. String motifs: Motifs such as CXXCXXC will match any substring that has first, fourth and last characters as C. 'X' denotes a wild card residue that can match any amino acid or nucleotide.
2. Range motifs: Motifs such as SEK{2,5}XXC would match SEKKAEC, SEKKKAEC... SEKKKKKAEC.
3. Either/or motifs: Certain amino acid residues or nucleotides in motifs can be specified using the '|' operator. For instance, AA(B|C)DE will match AABDE as well as AACDE. B and C can also be replaced by complex motifs to form a motif of the form AA(SEKXXAF)|(SEKP{2,4}DFX)DE.
4. Start of Sequence motifs: If the motifs are prefixed by ^, the match will be performed at the start of the sequence. Example, ^CDG will match only a CDG occurring at the start of the sequence and nowhere else in the string.
5. End of Sequence motifs: The motifs that are suffixed by $ will be matched only at the end of the sequence.
6. Class motifs: For motifs in which amino acid residues or nucleotides are enclosed in square brackets, the method will match any of them in any order against the sequence. For example, ABC[EFGH] will match ABCEFGH, ABCEFHG, ABCE, ABCGH etc.
7. Negative class motifs: If ^ is prefixed to the characters that are inside the [ ], SSMBS will ignore all matches of substrings that have the characters placed in [ ]. For example, ABC[^EFGH] will match ABC, ABCD but not ABCE i.e. all substrings beginning with ABC and not ending with E or F or G or H.
8. Multiple motifs can also be combined to form a single motif and searched accordingly. For instance, a motif of the form AATAX{3,10}GACATTX{20,30} TCACTG will attempt to match three smaller motifs in the order $motif_1$=AA TAX, $motif_2$=GACATT and $motif_3$=TCACTG such that 3-10 nucleotides separate $motif_1$ and $motif_2$, and 20-30 nucleotides separate $motif_2$ and $motif_3$.
9. Motifs with hydrophobic or polar residues: Hydrophobic or polar residues can be substituted by the single characters B or Z respectively.

3.2 Case Study: Members of the Haloacid Halogenase HAD Family

Based on the presence of three sequentially separated motifs, DXXX(V|T), (S|T)GX, KX_{16-30}(G|S)(D|S)XXX(D|N), protein sequences can be categorized

to belong to the HAD family of proteins [10],[7]. Thus, the proposed method was executed over a set of 15 protein sequences that belong to the enzyme trehalose-6-phosphatase. A sample of the output generated by SSMBS is shown below.

```
------------------------------------------------------------------------
Input FileName              :   fasta.txt
No of motifs to be searched :   3
Motif 1                     :   DXXX(V|T)
Motif 2                     :   (S|T)GX
Motif 3                     :   KX{16,30}(G|S)(D|S)XXX(D|N)
OutPut FileName             :   filename1.doc
------------------------------------------------------------------------
OUTPUT OF SSMBS
------------------------------------------------------------------------
>1L6R:A|PDBID|CHAIN|SEQUENCE
Motif positions for occurrence number: 1
(DGNLT: 13 to 17)
(SGN: 45 to 47)
(KAFAVNKLKEMYSLEYDEILVIGDSNND: 154 to 181)
Position of motifs with intermediate residues for occurrence number: 1
(DGNLT: 13 to 17)
(DRDRLISTKAIESIRSAEKKGLTVSLL)
(SGN: 45 to 47)
(VIPVVYALKIFLGINGPVFGENGGIMFDNDGSIKKFFSNEGTNKFLEEMSKRTSMRSILTNRWREASTG
FDIDPEDVDYVRKEAESRGFVIFYSGYSWHLMNRGED)
(KAFAVNKLKEMYSLEYDEILVIGDSNND: 154 to 181)
------------------------------------------------------------------------
Motif positions for occurrence number: 2
(DGNLT: 13 to 17)
(TGF: 115 to 117)
(KAFAVNKLKEMYSLEYDEILVIGDSNND: 154 to 181)
Position of motifs with intermediate residues for occurrence number: 2
(DGNLT: 13 to 17)
(DRDRLISTKAIESIRSAEKKGLTVSLLSGNVIPVVYALKIFLGINGPVFGENGGIMFDNDGSIKKFFSN
EGTNKFLEEMSKRTSMRSILTNRWREAS)
(TGF: 115 to 117)
(DIDPEDVDYVRKEAESRGFVIFYSGYSWHLMNRGED)
(KAFAVNKLKEMYSLEYDEILVIGDSNND: 154 to 181)
------------------------------------------------------------------------

------------------------------------------------------------------------
Total number of occurrences: 5
------------------------------------------------------------------------
>1L6R:B|PDBID|CHAIN|SEQUENCE
Motif positions for occurrence number: 1
(DGNLT: 13 to 17)
(SGN: 45 to 47)
(KAFAVNKLKEMYSLEYDEILVIGDSNND: 154 to 181)
Position of motifs with intermediate residues for occurrence number: 1
(DGNLT: 13 to 17)
```

```
(DRDRLISTKAIESIRSAEKKGLTVSLL)
(SGN: 45 to 47)
(VIPVVYALKIFLGINGPVFGENGGIMFDNDGSIKKFFSNEGTNKFLEEMSKRTSMRSILTNRWREASTG
FDIDPEDVDYVRKEAESRGFVIFYSGYSWHLMNRGED)
(KAFAVNKLKEMYSLEYDEILVIGDSNND: 154 to 181)
----------------------------------------------------------------------
**********************************************************************
93 hits were found in 15 sequences.
**********************************************************************
----------------------------------------------------------------------
Total sequences in file         :   15
Running on machine              :   igraph9.physics.iisc.ernet.in
Program stated at (h:m:s:ms)    :   3:40:6:559 on 2/3/07 3:40 AM
Program stop at   (h:m:s:ms)    :   3:40:6:994 on 2/3/07 3:40 AM
Executed Time     (h:m:s:ms)    :   0:0:0:435
----------------------------------------------------------------------
```

The output reports occurrences of the three motifs in each of the 15 sequences in the particular order as specified. This is in accordance with the results published in literature [12]. Hence, it can be concluded that all of the 15 sequences belong to the HAD family.

This test, however, could not be run directly on the PIR pattern search as three different motifs are be specified simultaneously, for which there is no provision on the web-server. On checking PROSITE for HAD, haloacid halogenase and combinations thereof, no signature motifs were found that could be used to provide a pattern to the PIR search.

3.3 Case Study: Transcription Activation of CRP in *E.coli*

The proposed method was tested to run over the genome sequence of *E.coli* to locate the occurrences of the CRP binding complex. According to the literature [2], the consensus for the activating regions of the CRP protein is given by the sequence S_1 = TGTGAX{5,7}TCACA. The whole complex inclusive of the CRP with the core promoter sites is specified by the consensus sequence S_2 = TGTGAX{5,7}TCACAX{15,23}TATAA [2]. SSMBS located 28 identical matching occurrences of S_1 and a single identical occurrence of S_2 in the genome sequence. A section of the output for S_2 search is shown below.

```
----------------------------------------------------------------------
No of motifs to be searched :   1
Motif 1             :   TGTGAX{5,7}TCACAX{15,23}TATAA
----------------------------------------------------------------------
OUTPUT OF SSMBS
----------------------------------------------------------------------
>gi|49175990|ref|NC_000913.2| Escherichia coli K12, complete genome
Motif positions for occurrence number: 1
...CAATCTTTA
(TGTGATACAAATCACATAAATACCCCTTTAATGTTATAA: 1986066 to 1986104)
AAATGATAAT...
```

```
**********************************************************************
1 hit(s) was found in 1 sequence(s).
**********************************************************************
Running on machine              :   igraph9.physics.iisc.ernet.in
Program stated at (h:m:s:ms)    :   21:39:12:101 on 2/6/07 9:39 PM
Program stop at   (h:m:s:ms)    :   21:39:13:585 on 2/6/07 9:39 PM
Executed Time     (h:m:s:ms)    :   0:0:1:484
----------------------------------------------------------------------
```

3.4 Case Study: Zinc Finger Binding Motif

In order to compare SSMBS with with the PIR Pattern Search in terms of speed and accuracy, the extremely well knowm Zinc Finger Binding Motif HX_3 HX_{23} CXXC was considered. SSMBS was used to search for this motif in the 90% non-redundant dataset of PDB chains containing 14,423 chains. It found 33 hits in 33 sequences in 7 seconds. A section of the output for the search from SSMBS is shown below.

```
----------------------------------------------------------------------
No of motifs to be searched :   1
Motif 1             :   HX{3}HX{23}CXXC
----------------------------------------------------------------------
OUTPUT OF SSMAS
----------------------------------------------------------------------
>1jrx_B mol:protein length:571     Flavocytochrome C
Motif positions for occurrence number: 1
...EVAETTKHE(HYNAHASHFPGEVACTSCHSAHEKSMVYCDSC: 54 to 85)HSFDFNMPYA...
----------------------------------------------------------------------
Total number of occurrences: 1
----------------------------------------------------------------------
>1wjd_B mol:protein length:55     Hiv-1 Integrase
Motif positions for occurrence number: 1
...DGIDKAQEE(HEKYHSNWRAMASDFNLPPVVAKEIVASCDKC: 12 to 43)QLKGEAMHGQ...
----------------------------------------------------------------------

----------------------------------------------------------------------
Total number of occurrences: 1
----------------------------------------------------------------------
**********************************************************************
33 hits were found in 33 sequences.
**********************************************************************
----------------------------------------------------------------------
Running on machine              :   igraph9.physics.iisc.ernet.in
Program stated at (h:m:s:ms)    :   7:28:58:273 on 6/14/08 7:29 AM
Program stop at   (h:m:s:ms)    :   7:29:5:279 on 6/14/08 7:29 AM
Executed Time     (h:m:s:ms)    :   0:0:7:6
----------------------------------------------------------------------
```

3.5 Case Study: Eukaryotic DNA Topoisomerase II

Following the results of the previous case study, another was carried out with the signature motif of the eukaryokic DNA Topoisomerase II protein: (L|I|V|M|A) R_{0-1} EG(D|N)SAF_{0-1} (S|T|A|G). A single sample output is shown, where the source file is the same as the earlier case study.

```
----------------------------------------------------------------------
No of motifs to be searched :   1
Motif 1             :   (L|I|V|M|A)R{0,1}EG(D|N)SAF{0,1}(S|T|A|G)
----------------------------------------------------------------------
OUTPUT OF SSMAS
----------------------------------------------------------------------
>1z0w_A mol:protein length:207     Putative Protease La Homolog Type
Motif positions for occurrence number: 1
...IQFVGTYEG(VEGDSAS: 91 to 97)ISIATAVISA...
----------------------------------------------------------------------
Total number of occurrences: 1
----------------------------------------------------------------------
**********************************************************************
25 hits were found in 25 sequences.
**********************************************************************
----------------------------------------------------------------------
Running on machine              :   igraph9.physics.iisc.ernet.in
Program stated at (h:m:s:ms)    :   8:6:56:702 on 6/14/08 8:07 AM
Program stop at   (h:m:s:ms)    :   8:7:4:77 on 6/14/08 8:07 AM
Executed Time     (h:m:s:ms)    :   0:0:8:375
----------------------------------------------------------------------
```

The same searches for the last two case studies (outlined in sections 3.4 and 3.5) performed by the PIR pattern search over the the UniRef100 database (with its several thousand sequences) timed out after 43 and 22 minutes respectively. This search was not attempted for *E. coli* genome case study since PIR pattern search cannot be used for nucleotides. The web-server has the additional disadvantage of depending upon the internet connectivity of the user, rather than being freely available and utilized. Thus, for simple common motif searches over large databases, perhaps the SSMBS algorithm is easier to use.

4 Implementation

SSMBS requires three input: a file of protein or nucleotide sequences in FASTA format, the number of motifs to be searched and the motifs of interest. The program will generate a detailed output containing the location of the motifs and the residues which separate the motifs occurring in the given order. An option is also provided to the user to specify the maximum number of occurrences to be reported per sequence. This is particularly helpful in case this method reports a large number of occurrences for the specified motifs. The number of motifs that can be detected in a particular sequence is restricted to five due to

the high time complexity of the method for more motifs. A standalone version of SSMBS can be obtained upon request by sending an e-mail to Dr. K. Sekar (sekar@serc.iisc.ernet.in or sekar@physics.iisc.ernet.in). We plan to create a web-based computing server to locate the sequentially separated motifs in various biological sequence databases such as SWISS-PROT, PDB, PIR and Genome Database.

The SSMBS method has been implemented using JAVA since it has an in-built garbage collector that works with commendable efficiency. It improves the performance of the program by releasing occupied portions of the memory that are no more in use during run time. Since JAVA is also a platform independent language, the program can be executed on any operating system. The program has been successfully tested on Microsoft Windows (XP), Linux (Red Hat 9.0) and Sun Solaris.

5 Conclusion

Sequentially Separated Motifs in Biological Sequences (SSMBS) is a motif localization method used to locate user-defined motifs in both nucleotide and protein sequences. It has been developed to provide a comprehensive solution to the task of locating sequence motifs occurring in a particular order in large biological sequence databases. The method also provides the option for the user to specify motifs using regular expressions. By default, the method locates all the overlapping occurrences of the motifs. The method has the advantage of locating the ordered occurrences of up to five motifs in any user-defined database in FASTA format. It is a rapid method and clearly indicates the location and occurrence of the motifs.

Acknowledgments. The authors gratefully acknowledge the use of the Bioinformatics Centre, the Interactive Graphics Based Molecular Modeling facility and the Supercomputer Education and Research Centre. The methodology presented here is supported by a research grant provided by the Department of Information Technology, Governement of India. Part of this work is supported by the Institute-wide computational biology programme.

References

1. Hulo, N., Sigrist, C.J.A., Bairoch, A.: Recent improvements to the PROSITE database. Nucl. Acids Res. 32, D134–D137 (2004)
2. Carvalho, A.M., Freitas, A.T., Oliveira, A.L., Sagot, M.: An Efficient Algorithm for the Identification of Structured Motifs in DNA Promoter Sequences. IEEE/ACM Transactions on Computational Biology and Bioinformatics 03, 126–140 (2006)
3. Cartharius, K., Frech, K., Grote, K., Klocke, B., Haltmeier, M., Klingenhoff, A., Frisch, M., Bayerlein, M., Werner, T.: MatInspector and beyond: promoter analysis based on transcription factor binding sites. Bioinformatics 21, 2933–2942 (2005)

4. Wingender, E., Chen, X., Fricke, E., Geffers, R., Hehl, R., Liebich, I., Krull, M., Matys, V., Michael, H., Ohnhaeuser, R., Prueb, M., Schacherer, F., Thiele, S., Urbach, S.: Match - a tool for searching transcription factor binding sites in DNA sequences. Nucl. Acids Res. 29, 281–283 (2001)
5. Akiyama, Y.: TFSEARCH: Searching Transcription Factor Binding Sites, http://www.rwcp.or.jp/papia/
6. Werner, T.: Model for prediction and recognition of eukaryotic promoters. Mammalian Genome 10, 168–175 (1999)
7. Wang, W., Kim, R., Jancarik, J., Yokota, H., Kim, S.H.: Crystal structure of phosphoserine phosphatase from Methanococcus jannaschii, a hyperthermophile, at 1.8 A resolution. Structure 9, 65–71 (2001)
8. VanHelden, J., André, B., Collado-Vides, J.: Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies. J. Mol. Biol. 281, 827–842 (1998)
9. Pavlidis, P., Furey, T.S., Liberto, M., Haussler, D., Grundy, W.N.: Promoter region-based classification of genes. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 151–163 (2001)
10. Collet, J.F., Stroobant, V., Pirard, M., Delpierre, G., Schaftingen, E.V.: A new class of phosphotransferases phosphorylated on an aspartate residue in an amino-terminal (DXDX(T/V)) motif. J. Biol. Chem. 273, 14107–14112
11. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. 147, 195–197 (1981)
12. Rao, K.N., Kumaran, D., Swaminathan, S.: Crystal structure of trehalose-6-phosphate phosphatase-related protein: Biochemical and biological implications. Protein Sci. 15, 1735–1744 (2006)
13. Altschul, S.F., Gish, W., Miller, W., Myers, W.E., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215, 403–410 (1990)
14. Nevill-Manning, C.G., Wu, T.D., Brutlag, D.L.: Highly specific protein sequence motifs for genome analysis. JOURNAL NAME HERE 95, 5865–5871 (1998)
15. Ben-Hur, A., Brutlag, D.: Remote homology detection: a motif based approach. Bioinformatics 19, i26–i33 (2003)
16. Russ Overbeek: scan_for_matches, http://iubio.bio.indiana.edu/soft/molbio/pattern/scan_for_matches
17. Dsouza, M., Larsen, N., Overbeek, R.: Searching for patterns in genomic data. Trends Genet. 13, 497–504 (1997)
18. Pesole, S., Liuni, S., D'Souza, M.: PatSearch: a pattern matcher software that finds functional elements in nucleotide and protein sequences and assesses their statistical significance. Bioinformatics 16, 439–450 (2000)
19. Huang, J.Y., Brutlag, S.: The eMOTIF Database. Nucl. Acids Res. 29, 202–204 (2001)
20. Obenauer, J.C., Cantley, L.C., Yaffe, M.B.: Scansite 2.0 Proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucl. Acids Res. 31, 3635–3641 (2003)
21. MOTIF SCAN, http://myhits.isb-sib.ch/cgi-bin/motif_scan
22. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L.L., Studholme, D.J., Yeats, C., Eddy, S.R.: The Pfam protein families database. Nucl. Acids Res. 32, D138–D141 (2004)