Extracting Biologically Relevant Common Motifs from Protein Sequences

Jack Nansheng Chen
MSS/ICICS
University of British Columbia
nchen@mss.cicsr.ubc.ca

Xiaohong Zhao
Department of Computer Science
University of British Columbia
xhzhao@cs.ubc.ca

Jack Xiang-Jia Min
MSS/ICICS
University of British Columbia
xmin@mss.cicsr.ubc.ca

Jason Xun Zhang [1]
MSS/ICICS
University of British Columbia
xuzhang@mss.cicsr.ubc.ca

April 1, 2002, *revised* April 14, 2002

Abstract

We have developed a novel algorithm that is capable of extracting biologically relevant common motifs from multiple protein sequences. A program, called MotifScan, was developed using this algorithm and then it can be pipelined with the widely used programs ClustalW and ProScan to facilitate motif extracting. MotifScan takes aligned multiple protein sequences processed using ClustalW as input, scans and scores these aligned sequences site-by-site according the well-established scoring matrix BLOSUM50 to extract "common sequences" shared by all input sequences. These common sequences are then fed into ProScan to obtain common motifs, annotated with biological functions according to data in ProSite. We have also demonstrated that our program outperforms the well-known program for extracting common motifs, MEME, in a number of ways, including time complexity, and insertion/deletion sensitivity.

1 Introduction

With the accomplishment of the Human Genome Project and the sequenced genomes of an increasing number of organisms being available, a fundamental issue in biological sequence analysis is the identification of sequence motifs as a means of suggesting good candidates for biologically functional regions such as promoters, splicing sites, binding

[1] Jason Xun Zhang audits this course (CPSC536a).

sites, phosphorylation sites, glycosylation sites, interaction sites, etc. The underlying reason for doing this is that same motifs are repeatedly used in different biological macromolecules such as proteins, DNA and RNA to carry out similar functional roles (Bertone and Gerstein, 2001).

Identification of sequence motifs is not only significant for annotating new genes, it is also useful in mining their new "cousins", sequences with a similar array of motifs. Since the initial effort of sequencing biological macromolecules (proteins, DNAs and RNAs) several dozens of years ago, intensive efforts have been made in developing algorithms and tools to search for functional (and structural) motifs. With the improvement of computer processing power and the wide spread use of personal computers, a number of new programs dealing with motif searching and especially related problems such as database "blasting" and sequence alignment have been developed.

Representative programs for searching for homologous sequences include BLAST (Altschul, Gish, Miller, Meyers and Lipman, 1990), FASTA (Lipman and Pearson, 1985), and more recently, PSI-BLAST (Altschul et al., 1997), which deals with both gaps and handles sequences with both conserved and non-conserved regions, but mainly used for extracting homologous sequences from databases. ClustalW (Thompson, Higgins, Gibson, 1994) has been developed to align multiple protein sequences. All of these programs have been in existence for over a decade and highly fine-tuned. ProScan (Prosite scan) is a program used to check for structural and functional motifs residing in protein sequences (Bairoch A, P. Bucher, K. Hofmann. 1997). Other programs designed to look for motifs include eMotif (Nevill-Manning, Wu, and Brutlag, 1998) and Pratt (Jonassen, Collins and Higgins, 1995). ProScan compares a given protein sequence against a motif database, Prosite. The program has received tremendous attention since its initial launch.

On the other hand, there is no easy way to extract common motifs from an array of protein sequences. The only program we have found widely used so far is one named MEME (Multiple EM for Motif Elicitation) developed by a group over four years at the San Diego Supercomputer Center (SDSC)(Bailey and Elkan, 1994). MEME is based on an algorithm called MM, which given a dataset of unaligned, possibly unrelated biopolymer sequences, estimates the parameters of a probabilistic model. The MM algorithm is an extension of the expectation maximization technique for fitting finite mixture models. MEME can find one or more motifs in a collection of biological sequences. The major drawback of MEME is that (1) it has a approximate time complexity $O(N^5)$ (for details, see later discussion) and therefore it is slow (need to wait hours even days, in case of longer sequences, for results after submission of datasets to the server at UCSD Supercomputer Center); (2) it doesn't handle insertions and deletions (gaps) within motifs (more details will be covered later).

We have proposed and implemented a novel program that can be used to rapidly extract common motifs from a collection of protein sequences and it is capable of handling insertion and deletion issues (gaps in motifs). Our program (MotifScan) is further

"pipelined" with existing programs ClustalW and ProScan to extract biologically relevant motifs (not just common subsequences) from a set of input protein sequences.

2 Project Design

Our project takes a collection of protein sequences as input, and returns sequences with common motifs "annotated" with biologically relevant data (motifs with experimentally verified functions). It consists of three major parts (Figure 1).

The **central part** of the project is a novel program, which is implemented by us and called "**MotifScan**". MotifScan differs from other motif finding programs in important ways. First of all, it takes advantage of existing programs for aligning multiple sequences, rather than aligning multiple sequences itself (MEME does not handle multiple sequence alignment) or taking alignment data from existing database, therefore it is more flexible (eMotif relies on the BLOCKS database, Nevill-Manning et al., 1998).

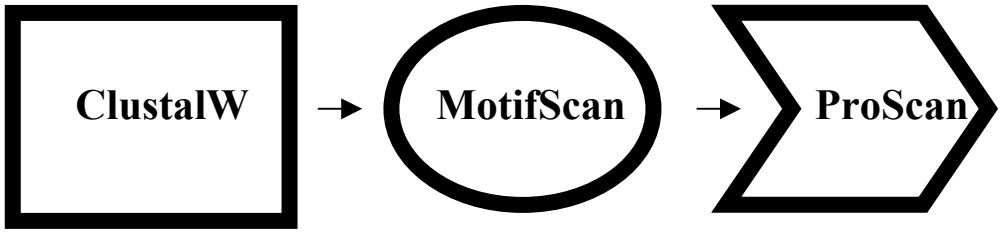

Figure 1 Overall design

MotifScan takes an array of pre-aligned protein sequences generated by ClustalW as its input, and returns all of the common motifs residing among these sequences as its output.

Input: a set of pre-aligned protein sequences.

Output: a set of common "motifs", which are subsequences shared by input protein sequences.

The **next part** of the project is to integrate ClustalW with **MotifScan**. Since ClustalW has been in existence over a decade and has been highly fine-tuned over time by extensive usage, we take the advantage of the program and use it to align input sequences and pass the aligned sequences into MotifScan. We anticipate that by doing so we can achieve best possible efficiency and reliability.

Input: a collection of protein sequences in FASTA format.

Output: a set of aligned protein sequences.

The **last part** of our project takes output from MotifScan as input and produces "functional" motifs as output. We integrate the program ProScan into our system for the job. The idea is that we can allow users of this program to view "meaningful" motifs after

they run our program. ProScan searches the "common motifs" found by MotifScan against the Prosite database, a collection of motifs that have been experimentally studied.

Input: a set of common "motifs".

Output: a set of "motifs" annotated with biological functions.

3 Algorithm of MotifScan

MotifScan reads in the aligned sequence data, and outputs the possible motifs with sequence name (in which the motif resides), motif name, start position, and length.

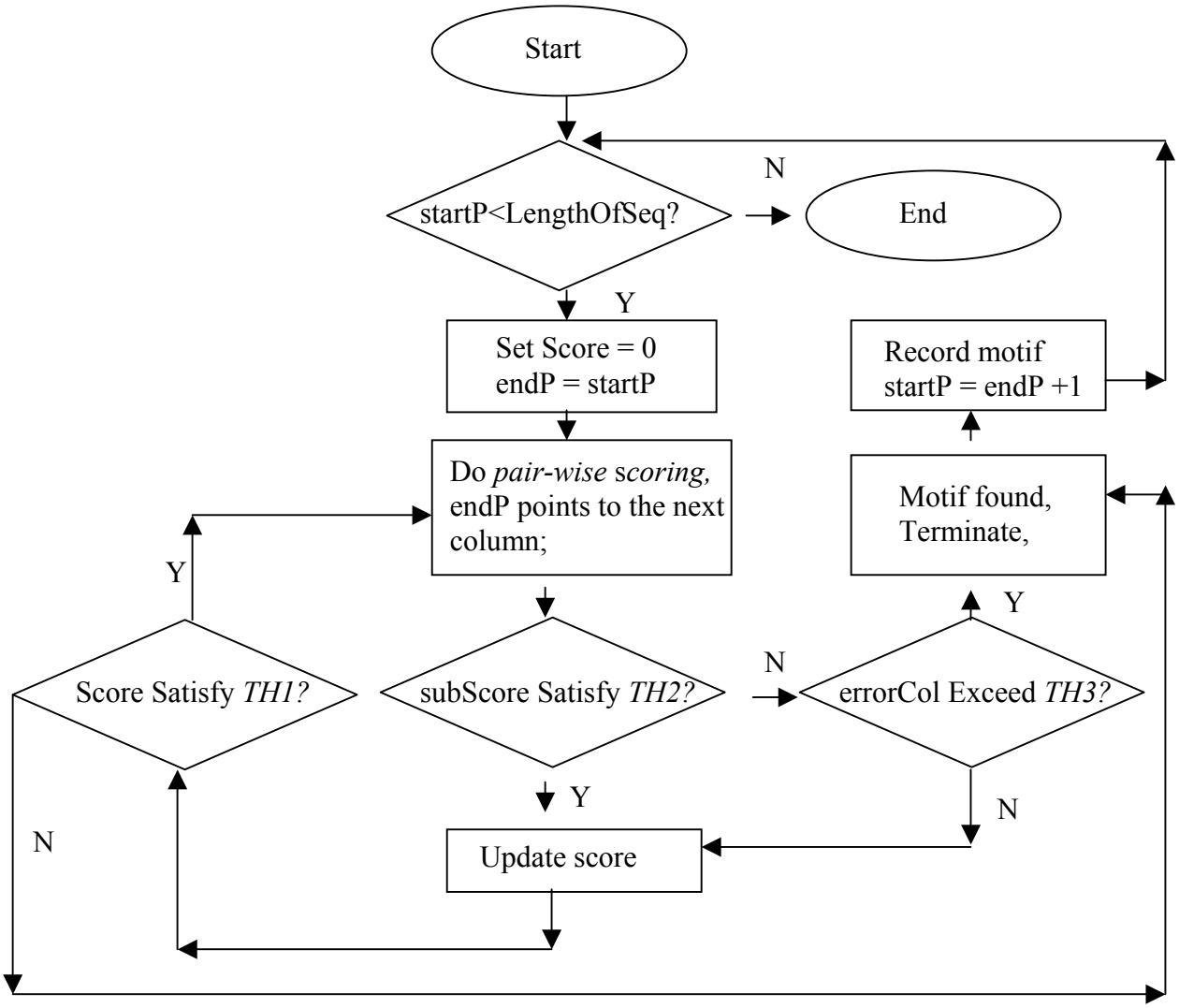

Figure 2. Illustration of algorithm used in MotifScan

Algorithm (Figure 2)

- Read in the aligned sequences from the output of the ClustalW (in .phy format). We've put them into a two-dimensional dynamic array with row equals the number of sequences, and column equals the length of the maximal sequence.

- Scoring matrix: We used a modified BLOSUM50 for calculating score.
- Setting the thresholds: Three thresholds are used for localizing motifs:

  TH1: The sum of the "motif block"(columns between the start position and the end position of a possible motif, including all sequences).

  TH2: The sum of the current column (of one corresponding site for all sequences).

  TH3: The number of "error columns" can be tolerated before termination of a motif (if qualified) and initialization of a new motif. For the purpose to avoid split of a motif into several short motifs, our program allow a user to set the number of columns with lower value of TH3.

- Searching for motifs. At the beginning, a start Pointer "startP" and an end Pointer "endP" are set to point to 0 respectively. Then follow the following steps:
  1. If "startP" is less than "col" (total sequence length, equal for all of the aligned sequences), calculate the total column score by adding all pair-wise scores of the column pointed by the "endP". Then set the "subSum" equal to the score.
  2. If "subSum" satisfies the TH2 (threshold 2), and the sum of current "motif block" satisfies the TH1 (threshold 1), do the following steps:
     I. Let sum = sum + subSum,
     II. Increase the endP by 1, and go to step 1,

     Otherwise, if "subSum" doesn't satisfy the TH2,

     If error columns hasn't exceeded the threshold 3,

     Record the number of error columns, and go to step 1.

     Else

     Go to step 3.
  3. If the length of the "motif block" is larger or equal to 3,
     I. Delete the trailing amino acids, whose sum doesn't satisfy the threshold 2. Since we can take in certain number of error columns, which is to allow some error in between the motifs. But if these errors appeared at the end of the possible motif, it's not meaningful to have them. So we delete them.
     II. Record the motif. And put it into a dynamic linked list.
     III. Reset the startP to 0, and goto step 1.

- Write the results into a file, which will be fed further into another program ProScan to get a set of common motifs with biological functions.

MotifScan is implemented in the C programming language. A Perl script was written to "pipleline" ClustalW, MotifScan and ProScan for ease of use.

4 Test Cases

Determining thresholds (TH1, TH2, and TH3)

All three thresholds have to be determined to run our program (MotifScan needs them). We have taken advantage of the fact that motifs for some protein families are well characterized and known. After extensive tuning, we arrived at an optimal value for each of these variables.

(1) TH1 = Number of sequences X 5;
(2) TH2 = Number of sequences X 1;
(3) TH3 = 2.

The datasets:

We've tested our program (focusing on **MotifScan)** using the lipocalin proteins and other datasets, which include a group of six proteins from Assignment 2 (for multiple alignment), and a group of four proteins belongs to NMDA receptor subfamily of glutamate receptor family (Ghosa, 2002). For all of the datasets, we have run both our programs and MEME. We are going to focus our report on lipocalin proteins because these sequences have been used to test MEME by the original developers (Bailey and Elkan, 1994).

```
     5    213
sp|P00305| ---------- ----GDIFYP GYCPDVKPVN DFDLSAFAGA WHEIAKLPLE
sp|P09464| MQYLIVLALV AAASANVYHD GACPEVKPVD NFDWSNYHGK WWEVAKYPNS
sp|P02754| ------MKCL LLALALTCGA QALIVTQTMK GLDIQKVAGT WYSLAMAASD
tr|Q61921  ------MLLL LCLGLTLVCV HAEEASSTGR NFNVEKINGE WHTIILAFDK
sp|P18902| ---------- -------ERD CRVSSFRVKE NFDKARFAGT WYAMAKKDPE

           NENQGKCTIA EYKYDGKKAS VYNSF-VSNG VKEYMEGDLE IAPDAKYTKQ
           VEKYGKCGWA EYTPEGKSVK VSNYH-VIHG KEYFIEG--- TAYPVGDSKI
           ISLLDAQSAP LRVYVEELKP TPEGD-LEIL LQKWENG--E CAQKKIIAEK
           REKIE-DNGN FRLFLEQIHV LENSL-VLKF HTVRDEE--- CSELSMVADK
           GLFLQDNIVA EFSVDENGHM SATAKGRVRL LNNWDV---- CADMVGTFTD

           GKYVMTFKFG QRVVNLVPW- ------VLAT DYKNYAINYN CDYHPDK-KA
           GKIYHKLTYG GVTKENVFN- ------VLST DNKNYIIGYY CKYDEDK-KG
           TKIPAVFKID ALNENKVL-- ------VLDT DYKKYLL--F CMENSAE-PE
           TEKAGEYSVT YDGFNTFT-- ------IPKT DYDNFLMAHL INENDGE-TF
           TEDPAKFKMK YWGVASFLQK GNDDHWIIDT DYETFAVQYS CRLLNLDGTC

           HSIHAWILSK SKV-LEGNTK EVVDNVLKTF SHLIDASKFI SNDFSEAACQ
           HQDFVWVLSR SKV-LTGEAK TAVENYLIG- SPVVDSQKLV YSDFSEAACK
           QSLACQCLVR TPE-VDDEAL EKFDKALKAL PMHIRLS--F NPTQLEEQCH
           QLMG--LYGR EPD-LSSDIK ERFAQLCEKH GILRENIIDL SNANRCLQAR
           ADSYSFVFAR DPSGFSPEVQ KIVRQRQEEL CLARQYRLIP HNGYCDGKSE

           YSTTYSLTGP DRH
           VNN------- ---
           I--------- ---
           E--------- ---
           RNIL------ ---
```

Figure 3 Proteins sequences dataset used to test our program MotfiScan

The lipocalin proteins bind small, hydrophobic ligands for a wide range of biological purposes. The data contains the five most divergent lipocalins with known 3D structure.

As mentioned earlier, MotifScan takes pre-aligned multiple protein sequences in .phy format (Figure 3), in which "5" indicates that there are five sequences to be processed and "213" indicates that there are a total 213 amino acids (including gaps) for each sequences.

```
pattern 1 from position 30 , length:16

[D|N|G][F|L][D|N][L|W|I|V|K][S|Q|E|A][A|N|K|R][F|Y|V|I][A|H|N]G[A|K|T|E]W[H|W|Y][E|S|T|A]
[I|V|L|M][A|I][K|M|L]

  ------------------------------------

 pattern 2 from position 109 , length:4
 [T|K|V|E][F|L|Y][K|T|S][F|Y|I|V|M]

  ------------------------------------

 pattern 3 from position 133 , length:8
 TD[Y|N][K|D|E][N|K|T][Y|F][A|I|L][I|L|M|V]

  ------------------------------------

 pattern 4 from position 170 , length:11
 [L|V|F][E|T|D|S][G|D|S|P][N|E|D][T|A|I|V][K|L|Q][E|T|K][V|A|K|R|I][V|F][D|E|A|R][N|K|Q]

  ------------------------------------

 pattern 5 from position 201 , length:4
 [E|C|D][A|E|L|G][A|Q|K][C|A|S]
```

Figure 4 Common motifs in regular expressions

Results from MotifScan:

The result of our pipelined program, common motifs from the five listed proteins (Figure 3), is displayed in Figure 5.

To illustrate how our pipelined program works, we listed the results from MotifScan below (Figure 4 and Table 1). Figure 4 lists all five common "intermediate motifs (IM)" found in these five input protein sequences in regular expression. An IM is a subsequence of a protein that potentially contains functional motifs. An IM is usually used interchangeably as a motif. Each IM has a serial name, a starting position, and a length. Characters in brackets are amino acid symbols. Actual "intermediate motifs" (IMs) are

listed in Table 1. We can see from Table 1 that IM1 starts at position 20 and terminates at position 46; IM2 starts at position 109 and terminates at position 113 and so on (Table 1).

ProScan finds motifs with biological functions

ProScan takes IMs and scans for motifs with biological functions assigned to it. Figure 5 lists results from ProScan. We can see that there is a common motif (LIPOCALIN **Lipocalin signature**) found for all of these sequences, and there is another motif (**phosphorylation site**). The results demonstrated that our algorithm has the ability to find the common motifs among different sequences.

Table 1 Common motifs obtained using MotifScan program (Lipocalin proteins)

	I	II	III	IV	V
	30 46	109 113	133 141	170 181	201 205
P00305	DFDLSAFAGAWHEIAK	TFKF	TDYKNYAI	LEGNTKEVVDN	EAAC
P09464	NFDWSNYHGKWWEVAK	KLTY	TDNKNYII	LTGEAKTAVEN	EAAC
P02754	GLDIQKVAGTWYSLAM	VFKI	TDYKKYLL	VDDEALEKFDK	EEQC
Q61921	NFNVEKINGEWHTIIL	EYSV	TDYDNFLM	LSSDIKERFAQ	CLQA
P18902	NFDKARFAGTWYAMAK	KFKM	TDYETFAV	FSPEVQKIVRQ	DGKS

>sp_P00305/1-14 : PS00213 LIPOCALIN **Lipocalin signature.**
DFDLSAFAGAWHEI
>sp_P09464/1-14 : PS00213 LIPOCALIN **Lipocalin signature**.
NFDWSNYHGKWWEV
>sp_P02754/9-14 : PS00008 MYRISTYL N-myristoylation site.
GTWYSL
>sp_P02754/1-14 : PS00213 LIPOCALIN **Lipocalin signature.**
GLDIQKVAGTWYSL
>tr_Q61921/1-16 : PS50025 LAM_G_DOMAIN L=-1 Laminin G domain profile.
NFNVEKINGEWHTIIL
>tr_Q61921/1-14 : PS00213 LIPOCALIN **Lipocalin signature**.
NFNVEKINGEWHTI
>sp_P18902/9-14 : PS00008 MYRISTYL N-myristoylation site.
GTWYAM
>sp_P18902/1-14 : PS00213 LIPOCALIN **Lipocalin signature**.
NFDKARFAGTWYAM

>sp_P00305/1-3 : PS00005 PKC_PHOSPHO_SITE Protein kinase C **phosphorylation site**.
TFK
>tr_Q61921/1-4 : PS00006 CK2_PHOSPHO_SITE Casein kinase II **phosphorylation site.**
TDYD
>sp_P18902/1-4 : PS00006 CK2_PHOSPHO_SITE Casein kinase II **phosphorylation site**.
TDYE
>sp_P09464/7-10 : PS00006 CK2_PHOSPHO_SITE Casein kinase II **phosphorylation site**.
TAVE

Figure 5 Output of Our pipelined program (ClustalW+MotifScan+ProScan)

Table 2 Common motifs obtained using MEME (Lipocalin proteins).

	I	II
P00305	14 PVN**DFDLSAFAGAWHEIAK** 33	103 PWVLA**TDYKNYAI**NYNC 120
P09464	18 PVD**NFDWSNYHGKWWEVAK** 37	114 FNVLS**TDNKNYII**GYYC 131
P02754	11 VKE**NFDKARFAGTWYAMAK** 30	104 HWIID**TDYETFAV**QYSC 121
Q61921	22 TMK**GLDIQKVAGTWYSLAM** 41	106 FTIPK**TDYDNFLM**AHLI 123
P18902	22 TGR**NFNVEKINGEWHTIIL** 41	108 VLVLD**TDYKKYLL**FCME 121

Results from MEME:

Motifs retrieved from these five proteins using MEME are listed in Table 2. It should be noted that the number of motifs retrieved by MEME is primarily determined by the user. MEME will use the MM algorithm to get the best candidates, without caring about the "quality" of the motifs. In this case, two motifs are requested and found. We requested two motifs, as stated by their authors when explain their method. Actually these two motifs are known before hand because of the known 3D structures. (Bailey and Alkan, 1994)

Another important point is that the position labeling is different between MEME result and our program, because we introduced gaps to achieve maximal alignment, while MEME doesn't.

MotifScan vs. MEME: a Comparison

To facilitate comparison, we have highlighted some amino acids in Table 1 in black. These black characters are motifs found using our programs (MotifScan), which indicate that the motifs found using MEME can basically be found using MotifScan. Actually MotifScan retrieves three more motifs (Table 1).

Our pipelined program can go a step further to retrieve functions for these found motifs (Figure 5). Our program declares that two functionally significant motifs are found existing in all of these five input protein sequences.

5 Discussion

MEME utilized the algorithm MM, which implements the technique of expectation maximization to fit a two-component finite mixture model to the set of sequences (as presented in the introduction). Multiple motifs are found by fitting a mixture model to the input data, probabilistically erasing the occurrences of the motif thus found, and repeating the process to find successive motifs (Bailey and Alkan, 1994). The algorithm does not guarantee to find the maximum likelihood estimates of the parameters of the model, only a local maximum. Also, different starting points (i.e., different initial values for the model parameters θ) can yield different solutions with varying likelihood values. It is usually necessary to run MM from several starting points, and pick the solution with the highest likelihood value. It's difficult to know when to stop (Bailey and Alkan, 1994). Though

MEME makes some improvements of the above weak points, the execution time of it is still approximately $O((NM)^2W)$, where N is the number of sequences in the dataset, M is their average length, W is the width of the motif. (Bailey and Alkan, 1994).

Our algorithm is theoretically straight forward, and the execution time of the MotifScan part is only $O(N^2M)$, where M is the length of each sequence which is identical after alignment, N is the number of sequences in the dataset. Nevertheless the performance of it is rather promising. We do not need to sample the data several times; one time of the scan can get the results. Our algorithm does not depend on the number of motifs a user specified (which is not meaningful when a user has no idea about how many motifs are likely to appear in the sequences), it would find any possible motifs among the dataset. Also, it doesn't depend on the starting points; the results are depending on three thresholds we've mentioned before.

The thresholds one (TH1) and two (TH2) are specified after we've done several experiments, and we find out that these thresholds are more suitable for wide range of datasets. The third threshold (the error columns we allow for the motifs) is actually can be modified to 3 as appropriate.

Since our program "calls" ClustalW for multiple alignment and ClustalW introduces gaps whenever necessary to promote maximal alignment, our program can retrieve motifs with necessarily inserted gaps, therefore it can detect motifs with insertion and/or deletion. On the other hand, MEME does not have the mechanism to handle insertion/deletion problems.

A detailed comparison is included in Table 3.

Table 3 Comparison in time execution of our program MotifScan and MEME[2]

	Our program	**MEME**
Complexity	MotifScan $O(N^2M)$	$O((NM)^2W)$
Handle insertion/deletion	Yes	No
Number of motifs	Determined by program	Determined by user (return any number of motifs requested by user)
Use existing programs	ClustalW, ProScan/Prosite	No

N: number of sequences; M: length of each sequence; W: motif length which is specified by the user.

Weakness of our program compared to MEME

[2] We haven't included the time complexity of the ClustalW part in our article. One reason is that there's no specific data indicated its time complexity in the related papers, and the other reason is that we consider by putting it in will not significantly affect the results we got from comparing the two algorithms.

A major limitation of our program is that it relies on ClustalW, which indicates that the input sequences should be "reasonably" similar. On the other hand, MEME does not have this limitation.

Concluding remarks and future work:

We have created a program MotifScan, which works with ClustalW and ProScan to retrieve common motifs from multiple protein sequences rapidly.
And one possible future work will be that we experiment with more datasets, and get the statistical results to show the difference between our algorithm and other existing algorithms.

6 Reference

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D. J., 1990. Basic local alignment search tool. *J. Mol. Biol.* 215: 403-410.
2. Altschul, S.F., Madden, T.L., Schaffer, A. A., Zhang J. Zhang, Z., Miller and Lipman D. J.,1997 Gapped BLAST and PSI-BLAST: a new generation of protein search programs. *Nucleic Acids Research*,25(17):3389-3402.
3. Bailey TL and Elkan C. 1994. Fitting a mixture model by expectation maximization to discove motifs in biopolymers. In *Proceedings of the second International Conference on Intelligent System for Molecular Biology.* pp 28-36. AAAI Press.
4. Bairoch A, P. Bucher, K. Hofmann. 1997. The PROSITE database, its status in 1997. Nucleic Acids Res. 25(1):217-221
5. Bertone and Gerstein, 2001. Yale University. The new direction in bioinformatics: integrative data mining for genomics and proteomics.
6. Ghosa, A. 2002. Neurobiology. Learning more about NMDA receptor regulation. Science. 2002 Jan 18, 295(5554):449-51
7. Lipman, D. J. and Pearson, W. R., 1985. Rapid and sensitive protein similarity searches. *Science* 227: 1435-1441.
8. Luscombe, Greenbaum and Gerstein, 2001. Yale University. What is bioinformatics? An introduction and overview.
9. Maddison, D. R., D. L. Swofford and W. P. Maddison. 1997. NEXUS: an extensible file format for systematic information. *Systematic Biology* 46:590-621.
10. Nevill-Manning, C.G., Wu, T.D. and Brutlag D.L., 1998. Highly specific protein sequence motifs for genome analysis. *Proc. Natl. Acad. Sci.* USA.
11. Thompson J.D., Higgins D.G., Gibson T.J.; 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22:4673-4680.

Appendix

Modified BLOSUM50

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-1	-1	-5
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3	-1	0	-1	-5
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3	4	0	-1	-5
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4	5	1	-1	-5
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	-3	-3	-2	-5
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3	0	4	-1	-5
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3	1	5	-1	-5
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4	-1	-2	-2	-5
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4	0	0	-1	-5
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4	-4	-3	-1	-5
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1	-4	-3	-1	-5
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3	0	1	-1	-5
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1	-3	-1	-1	-5
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1	-4	-4	-2	-5
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3	-2	-1	-2	-5
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2	0	0	-1	-5
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0	0	-1	0	-5
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3	-5	-2	-3	-5
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1	-3	-2	-1	-5
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	-4	-3	-1	-5
B	-2	-1	4	5	-3	0	1	-1	0	-4	-4	0	-3	-4	-2	0	0	-5	-3	-4	5	2	-1	-5
Z	-1	0	0	1	-3	4	5	-2	0	-3	-3	1	-1	-4	-1	0	-1	-2	-2	-3	2	5	-1	-5
X	-1	-1	-1	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-1	0	-3	-1	-1	-1	-1	-1	-5
*****	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5[3]

[3] Original BLOSUM50 assigns a score of 1 to gap-gap matches. To penalize gap-gap matches in our case, we modified BLOSUM50 and set this score to –5, same to a gap-"any amino acid" matche.