

# PMS: A Panoptic Motif Search Tool

Hieu Dinh, Sanguthevar Rajasekaran\*

Computer Science and Engineering Department, University of Connecticut, Storrs, Connecticut, United States of America

## Abstract

**Background:** Identification of DNA/Protein motifs is a crucial problem for biologists. Computational techniques could be of great help in this identification. In this direction, many computational models for motifs have been proposed in the literature.

**Methods:** One such important model is the  $(\ell, d)$  motif model. In this paper we describe a motif search web tool that predominantly employs this motif model. This web tool exploits the state-of-the-art algorithms for solving the  $(\ell, d)$  motif search problem.

**Results:** The online tool has been helping scientists identify many unknown motifs. Many of our predictions have been successfully verified as well. We hope that this paper will expose this crucial tool to many more scientists.

**Availability and requirements:** Project name: PMS - Panoptic Motif Search Tool. Project home page: <http://pms.engr.uconn.edu> or <http://motifsearch.com>. Licence: PMS tools will be readily available to any scientist wishing to use it for non-commercial purposes, without restrictions. The online tool is freely available without login.

**Citation:** Dinh H, Rajasekaran S (2013) PMS: A Panoptic Motif Search Tool. PLoS ONE 8(12): e80660. doi:10.1371/journal.pone.0080660

**Editor:** Gajendra P. S. Raghava, CSIR-Institute of Microbial Technology, India

**Received:** April 26, 2013; **Accepted:** October 6, 2013; **Published:** December 4, 2013

**Copyright:** © 2013 Dinh, Rajasekaran. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** This work has been supported in part by the following grants: NSF 0829916 and NIH R01-LM010101. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: [rajasek@engr.uconn.edu](mailto:rajasek@engr.uconn.edu)

## Introduction

Motif search is an important problem in biology. Computational techniques could greatly help in solving this problem. A number of computational motif search tools can be found in the literature. See e.g., PRATT [1], MEME [2], DILIMOT [3], SLIMDisc [4], SLIMFinder [5] and FIRE-pro [6].

Each of the above tools is based on a specific model of motif search. An important model for motifs is the  $(\ell, d)$ -motif search model. A simple version of this model can be stated as follows. We are given  $n$  input sequences  $s_1, s_2, \dots, s_n$  each of length  $m$ . Input are also two integers  $\ell$  and  $d$ . The problem is to find a motif  $M$  that is present in the  $n$  input sequences. It is known that  $M$  is of length  $\ell$  and that it occurs in each of the  $n$  input sequences within a Hamming distance of  $d$ .

This model has been shown to yield better sensitivities than that of other models when tested on known biological data (see e.g., [7]). The problem of  $(\ell, d)$ -motif search is intractable [8]. There are numerous algorithms that have been proposed for solving the  $(\ell, d)$ -motif search problem. Examples are RISO and RISOTTO [9]. But RISO and RISOTTO are down-loadable programs and there are no corresponding web systems. In this paper we describe a web system for motif search that uses the  $(\ell, d)$ -motif model. Our web system has the following features: 1) We employ several state-of-the-art algorithms for  $(\ell, d)$ -motif search. We can identify longer motifs than RISO and RISOTTO. RISO can only identify motifs of length up to 14. PMS can identify motifs of length up to 23; 2) Both DNA and protein motifs are supported; 3) We support quorum motif search. In this case the motif(s) need not occur in all

the input sequences. Quorum motif search is significantly more difficult than the regular version [10]; 4) Dyads motifs are also found. In particular, the dyad motif under concern could consist of two segments separated by a gap; 5) We employ a scoring mechanism for the putative motifs found; and 6) The user interface for PMS is very friendly; 7) In PMS, user emails are optional.

To the best of our knowledge, there is no other comprehensive motif search system, based on the  $(\ell, d)$ -motif model, comparable to ours.

## Results

### The PMS Webserver

The PMS server is freely available at <http://pms.engr.uconn.edu> or at <http://motifsearch.com>. The website is open to any user. Login is not required. However, any user with a login account will have the benefit of viewing and retrieving his or her submission(s) history. Also, a submission associated with a registered user will be kept in the system forever unless the user deletes it. Any submission from a user without a login account will be stored in the system for one month. It will be automatically removed after one month.

The purpose of the motif search tool is to help biologists identify novel motifs that may be present in input DNA and/or Protein sequences. Simple and user-friendly input forms will allow users to submit queries easily and quickly. Informative output and visualizations will permit users to analyze the results carefully. These features of the website are described in more detail in the following sections.

## Input Sequences and Parameters

The input data can be either DNA or protein sequences. The length of each sequence is required to be between 15 and 1000. The number of input sequences is required to be between 5 and 500. The input sequences should be organized in the well-known text-based format - FASTA.

For each input dataset, a set of parameters will be chosen by the user. These parameters are shown in Figure 1. The first parameter is called “*quorum percent*” which is the minimum percentage of the input sequences that contain motifs. Quorum percent is set to 75% by default.

The second parameter allows users to choose the structure of motifs. Currently, the tool considers two structures, namely, monads and dyads. A monad is a contiguous string and a dyad consists of two segments separated by a gap. A monad is assumed by default. For monads, the users will choose the motif length. By default, the motif length is chosen to be “Any” which means that the tool will search for motifs of lengths between 10 and 25. If information about the motif length is known, we recommend that it be used to reduce the processing time. For dyads, users should choose the length of the first segment or box, the length of the second box and the length of the gap between the two boxes. If the lengths are chosen to be “Any”, processing will proceed similar to that for monads.

The third parameter is for DNA sequences that allows users to have the option of considering the reverse complement sequences. If the input DNA sequences have the same orientation, the third parameter should be chosen to be “No”. Otherwise, we recommend that it be chosen to be “Yes”.

## Submitting Jobs

After entering the sequences and relevant parameters, the user clicks on the “Submit” button on the submission form. If the data entered are valid, the submission will enter the processing queue. Once the processing is over, a results page will be displayed. Information about the submission will appear on top of the results

page as shown in Figure 2. Users can update contact email or change the parameters by clicking on either the “Update” button or the “Change parameters” button, respectively.

After submission, the submission status could be one of these: in processing queue, being processed, and processed. If the submission has not been processed yet, the bottom of the results page will appear as shown in Figure 3. Users can click on the “Refresh” button to update the processing status. Users can either wait for their submission to be processed or bookmark the results page and return to it later. If the contact email is provided, the system will send a notification email when the submission is processed. The notification email will include the URL for the results page.

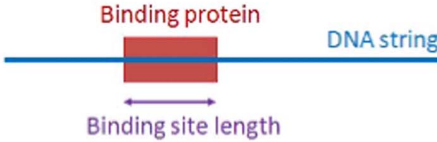
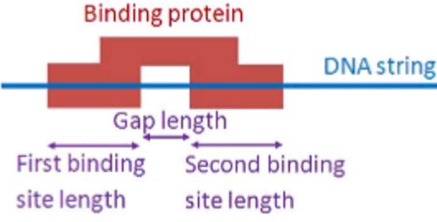
The processing time of any submission varies from a few minutes to a few hours, depending on the data, the parameters, and the workload of the server. If the user feels that the tool is taking too much time to process, we recommend that (s)he provide his/her contact email. Providing emails has a number of benefits. The first benefit is that the user will receive email notifications when query processing is complete. The second benefit is that their submissions will be stored in the system as long as they want. The third and perhaps the most important benefit is that they can retrieve their submission histories (as discussed in the next section).

## Output

Once the submission is processed, the bottom of the results page will appear as shown in Figure 4. Identified candidate motif(s) will appear on the left and the input sequences will appear on the right. If no motifs are found, we recommend to reduce the value of the quorum percent.

The candidate motif(s) found are ranked according to their scores. The score of a candidate motif is the logarithm of the probability that the motif occurs by random chance. The smaller the score, the more biologically significant the motif is. For more details on the scoring scheme, the readers are referred to [10]. For each candidate motif, users can click on the “View motif locations” button corresponding to the motif in order to view its

- The percentage of DNA sequences containing motifs \*(required):
- Please let us know about the form of motifs \*(required):
  - Single-box binding site
    - The length of the binding site:
  - Double-box binding site
    - The length of the first binding site:
    - The length of the gap between the two binding sites:
    - The length of the second binding site:

- Do you want to add the reverse complement sequences? \*(required):  Yes  No

**Figure 1. Parameters for DNA sequences.** The set of required parameters for DNA sequences. The first parameter is “*quorum percent*” which is the minimum percentage of the input sequences containing motifs. The second parameter allows users to choose the structure of motifs. doi:10.1371/journal.pone.0080660.g001

## Query Information

- **Query No:** 458
- **Contact email:** unknown
- **Single binding site:**
  - Binding site length: Any
- **Quorum Percent:** 75
- **Processing status:** being processed
- **Search mode:** Full Search
- **Submitted time:** 2012-03-01 17:08:36
- **Add reverse complement sequences:** No
- **Description:**

**Figure 2. Query information.** Information about submission. Users can click on the “Update” button or the “Change parameters” button to update the contact email or change parameters.  
doi:10.1371/journal.pone.0080660.g002

locations, i.e., its instances, in the input sequences. The locations of the motif instances will be highlighted in the input sequences as shown in Figure 4. The probability weight matrix of the motif is directly calculated through its motif instances and will appear above the input sequences. The probability of a DNA character at each column in the probability weight matrix is its frequency when its motif instances are aligned. When a motif is chosen, users can click on the “Save motif locations in text” button to save its locations in a text file.

For input protein sequences, the results are shown in Figure 5 which is similar to that of DNA sequences except that the probability weight matrix is not shown because it would be large for protein sequences.

### Submissions History

The website allows users to easily manage their submission(s) history. To start the submissions history feature, click on the link “Submission history” on the left menu of the website. To view submissions history, enter the contact email and password on the submissions history form. If the password has not been set by a user yet, (s)he can go to the reset password form and enter the contact email. An email will be sent to the contact email including a URL that allows the user to reset the password.

The list of submissions will be shown as in Figure 6. Users can sort their submissions based on query ID, submission time, or processing status. If the users want to view a particular submission, they can click on the link “View detail” of the corresponding submission.

### Feedback

The website supports an extensive feedback section. Users can easily submit feedbacks, comments, and questions using the feedback form. Feedbacks and comments will help us improve the website. To access the feedback form, click on the link “Feedback” on the left menu of the website.

### Discussion

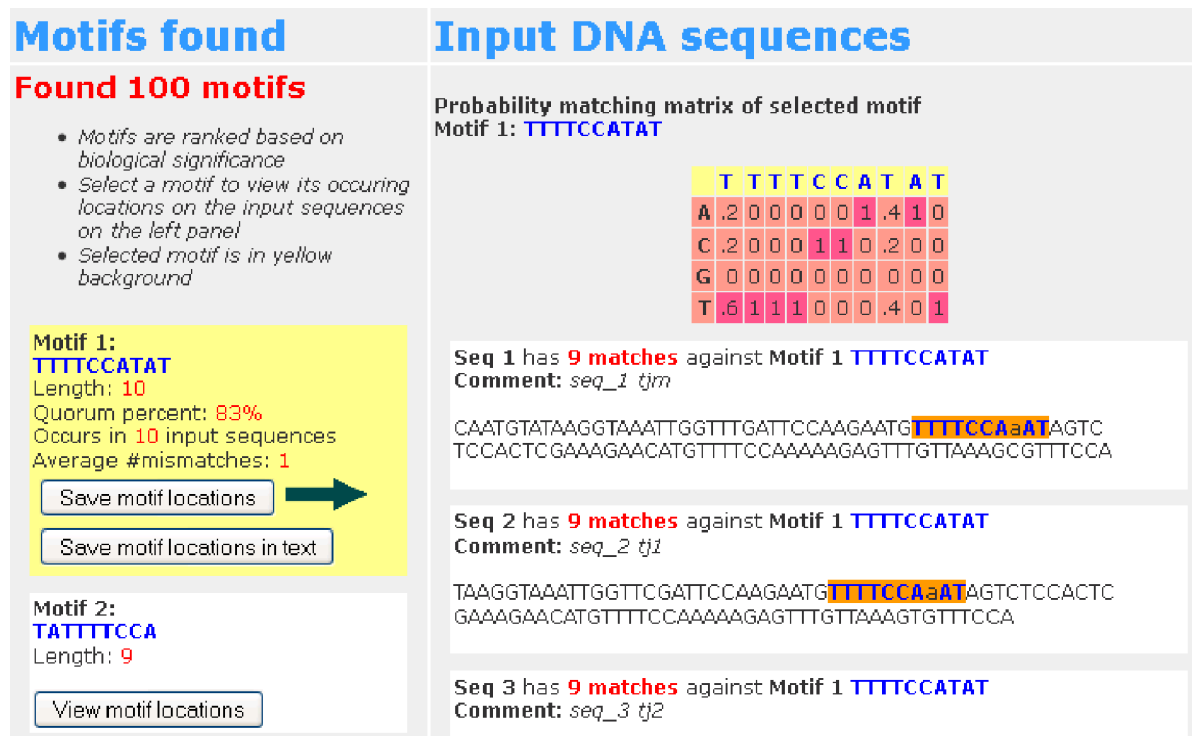
In this paper we have described a new web tool for motif search called PMS. This tool is based on the  $(\ell, d)$ -motif search model. This is a comprehensive web tool offering many crucial features and we are not aware of any other computational motif search tool comparable to ours. In future we plan to support additional features. For example, we will identify candidate motifs with more than two segments (separated by gaps). Another important feature will be to score the candidate motifs based on experimental data publicly available. User feedbacks will also be taken into account in enhancing the features of our web tool PMS. We also plan to incorporate other motif models in future. In addition we plan to work on finding longer motifs.

### Materials and Methods

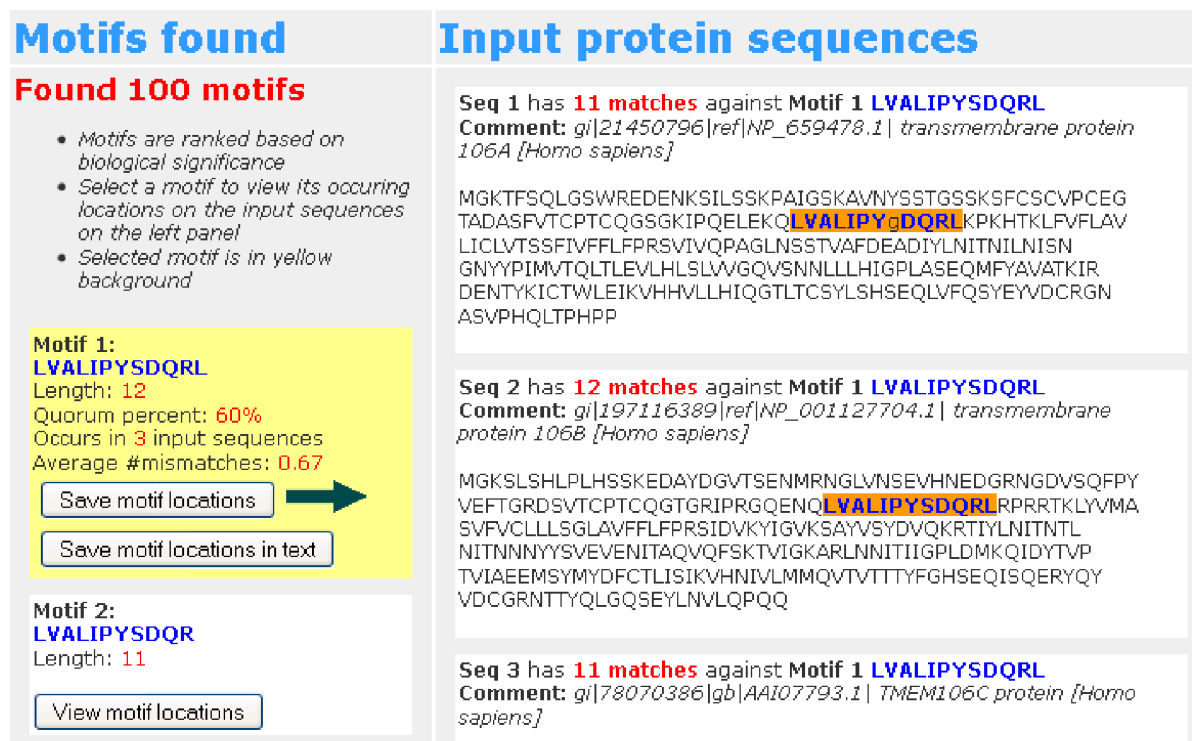
Our online motif search tool is built on state-of-the-art algorithms for the most well-known motif model -  $(\ell, d)$ -motif search or the Planted Motif Search (PMS). The PMS model has been shown to be very effective in identifying motifs (see e.g., [7]). The PMS Problem is defined as follows.

Motifs found	Input DNA sequences
<p>Not available. Please wait ...</p> <input type="button" value="Refresh"/> <p><i>You can save the link, close your web browser and check it back later. Or let us know your email by clicking the “Update” button. We will send you a notification email as soon as the result is available. The notification email will include the link to the result. Your email will be kept confidential. Thank you.</i></p>	<p><b>Seq 1</b></p> <pre>AC CCTCTTGTATAGCAGCAGATGACCGGGTGTG CCACTCATAGCCTTCC GATGGAGAGAAGCGCGGCCACTAGAAGATAATGTCGGGCCCTTGAGCGC GCC</pre> <p><b>Seq 2</b></p> <pre>TTGACTGGACTGCTCAAATCGATCAGGAATTTGCGCGCTAAAAACCTG GACGACAAACCGTTGCCAGGAGACGGTCTGACCCGTCTAATGAGCGACT ATAAGGTCGTCGCCACGGTATGTC CCAAGGGTAAAA CCACTGACCACTG GCCAATGAGAAAAGTAGATTGGCCACGTACTCGCCTCTCTCTGTCAGGAA CAACATAAGCCCGACCGG</pre> <p><b>Seq 3</b></p> <pre>AC CCTCTTGTATAGCAGCAGATGACCGGGTGTG CCACTCATAGCCTTCC GATGGAGAGAAGCGCGGCCACTAGAAGATAATGTCGGGCCCTTGAGCGC GCC</pre>

**Figure 3. Result not available.** An example of the results page when the submission has not been processed yet. Users can click on the “Refresh” button to update the processing status.  
doi:10.1371/journal.pone.0080660.g003



**Figure 4. Result available - DNA sequences.** An example of the results page when the submission is processed. The locations of the second motif are marked on the DNA input sequences.  
doi:10.1371/journal.pone.0080660.g004



**Figure 5. Result available - Protein sequences.** An example of the results page when the submission is processed. The locations of the second motif are marked on the protein input sequences.  
doi:10.1371/journal.pone.0080660.g005

Query No	Submission time	Data	Parameters	Description	Status	Result
↑ sort ↓	↑ sort ↓				↑ sort ↓	
427 <a href="#">View detail</a>	2012-02-12 17:45:41	Seq type: <b>Protein</b> #seqs: <b>6</b> Total length: <b>3526</b>	<b>Full Search</b> Quorum percent: <b>55</b> <b>Single-box</b> Length: <b>Any</b>		processed	#motifs: <b>100</b>
426 <a href="#">View detail</a>	2012-02-12 17:45:09	Seq type: <b>DNA</b> #seqs: <b>6</b> Total length: <b>584</b>	<b>Full Search</b> Quorum percent: <b>35</b> Add rvrs cmpltnt: <b>Yes</b> <b>Double-box</b> 1st length: <b>Any</b> gap length: <b>Any</b> 2nd length: <b>Any</b>		processed	#motifs: <b>100</b>
425 <a href="#">View detail</a>	2012-02-12 17:44:18	Seq type: <b>DNA</b> #seqs: <b>6</b> Total length: <b>584</b>	<b>Quick Search</b> Quorum percent: <b>75</b> Add rvrs cmpltnt: <b>Yes</b> <b>Single-box</b> Length: <b>Any</b>		processed	#motifs: <b>100</b>

**Figure 6. Submission history.** An example of the submissions history. Users can sort their submissions based on query ID, submission time, or processing status. Users can view a particular submission in detail by clicking on the link “View detail” of the according submission. doi:10.1371/journal.pone.0080660.g006

**Definition 0.1 PMS Problem:** given  $n$  sequences and integer parameters  $\ell, d$  and  $q$ , find all strings  $M$  of length  $\ell$  such that  $M$  appears in at least  $q$  out of the  $n$  given sequences within  $d$  mutations. Each such string  $M$  is a putative motif. Any  $\ell$ -mer (i.e., a substring of length  $\ell$ )  $L$  in any input string such that the Hamming distance between  $M$  and  $L$  is at most  $d$  is known as an instance of the motif  $M$ .

### The PMS Algorithms

In our web tool, we have used a combination of the current best PMS algorithms proposed in [10], [11], and [12].

We now summarize some of the techniques used in these algorithms.

Let  $HD(x, y)$  stand for the Hamming distance between two strings  $x$  and  $y$  of the same length. Let  $s_1, s_2, \dots, s_n$  be the given set of input sequences each of length  $m$ . For simplicity, consider the version where  $q = n$ . The PMS0 algorithm works as follows [13]: Consider  $s_1$ . Let  $L$  be an  $\ell$ -mer of  $s_1$ . Define the  $d$ -neighborhood  $B_d(L)$  of  $L$  to be the collection of all the  $\ell$ -mers  $q$  such that  $HD(L, q) \leq d$ . If  $L$  is an instance of an  $(\ell, d)$ -motif  $M$ , then, clearly  $M$  will be in  $B_d(L)$ . However, we do not know which  $\ell$ -mers of  $s_1$  are instances of the motif we are looking for. Thus, PMS0 constructs  $B_d(L)$  for every  $\ell$ -mer  $L$  in  $s_1$ . It then performs a union  $C_1$  of all of these  $d$ -neighborhoods.  $C_1$  contains all the  $(\ell, d)$ -motifs. For each  $\ell$ -mer  $L$  in  $C_1$ , the algorithm checks if  $L$  is an  $(\ell, d)$ -motif or not in an obvious manner. Note that for a given  $\ell$ -mer  $L$ , we check if it is an  $(\ell, d)$ -motif or not in  $O(mn\ell)$  time. A variation of this algorithm is called PMS1 and is described below [13]:

#### Algorithm PMS1

1. Compute  $C_i$  for each input sequence  $s_i$ ,  $1 \leq i \leq n$ . Here  $C_i = \bigcup_{L \in s_i} B_d(L)$ . In other words,  $C_i$  is nothing but the union

of  $d$ -neighborhoods of all the  $\ell$ -mers in  $s_i$ ,  $1 \leq i \leq n$ . The notation  $L \in s_i$  indicates that the  $\ell$ -mer  $L$  is a substring in  $s_i$ .

2. The  $(\ell, d)$ -motifs are now computed as  $\bigcap_{i=1}^n C_i$ .

Algorithm PMS5 can be thought of as an extension of PMS0 [11]. If  $S$  is a collection of strings, let  $M_\ell^d(S)$  denote the  $(\ell, d)$ -motifs present in  $S$ . If the input sequences are  $s_1, s_2, \dots, s_n$ , let  $S = \{s_1, s_2, \dots, s_n\}$  and let  $S' = \{s_2, s_3, \dots, s_n\}$ . The idea of PMS5 is to compute the  $(\ell, d)$ -motifs of  $S$  as  $\bigcup_{L \in s_1} M_\ell^d(L, S')$ .

In order to compute  $M_\ell^d(L, S')$  for any  $\ell$ -mer  $L$ , the algorithm uses a subroutine to compute the common  $d$ -neighborhood of three  $\ell$ -mers. Specifically, let  $x, y, z$  be any three  $\ell$ -mers. We use  $B_d(x, y, z)$  to denote the common  $d$ -neighborhood of  $x, y$ , and  $z$ . In other words,  $B_d(x, y, z)$  is nothing but the set of all  $\ell$ -mers that are at a distance of no more than  $d$  from each of the three  $\ell$ -mers  $x, y$ , and  $z$ .

To compute  $B_d(x, y, z)$ , PMS5 represents  $B_d(x)$  as a tree  $T_d(x)$ . Each node in this tree is an  $\ell$ -mer in  $B_d(x)$ . The root of  $T_d(x)$  is the  $\ell$ -mer  $x$ . The depth of  $T_d(x)$  is  $d$ .  $T_d(x)$  is traversed in a depth-first manner. Let  $t$  be any node in this tree. During the traversal,  $t$  will be output if  $t$  is in  $B_d(y) \cap B_d(z)$ . While visiting any node  $t$ , we check if there is a descendent  $t'$  of  $t$  such that  $t'$  is in  $B_d(y) \cap B_d(z)$ . The subtree rooted at  $t$  will be pruned if there is no such descendent. The problem of checking if  $t$  has any descendent that is in  $B_d(y) \cap B_d(z)$  is formulated as an integer linear program (ILP) on ten variables. This ILP is solved in  $O(1)$  time.

Any algorithm for solving the PMS problem when  $q \neq n$  is typically named with a prefix of ‘q’. One of the first algorithms to address this version of the PMS problem was qPMSPrune [12]. Algorithm qPMSPrune is based on the following observation: If  $M$  is any  $(\ell, d, q)$ -motif of the input strings  $s_1, \dots, s_n$ , then there exists an  $i$  (with  $1 \leq i \leq n - q + 1$ ) and an  $\ell$ -mer  $x \in s_i$  such that  $M$  is in

$B_d(x)$  and  $M$  is an  $(\ell, d, q-1)$ -motif of the input strings excluding  $s_i$ . The algorithm runs through every possible value of  $i$ ,  $1 \leq i \leq n$ . For a given value of  $i$ , it considers every  $\ell$ -mer  $x$  of  $s_i$ . Specifically, it constructs  $B_d(x)$  and identifies elements of  $B_d(x)$  that are  $(\ell, d, q-1)$  motifs (with respect to input strings other than  $s_i$ ).  $B_d(x)$  is represented as a tree with  $x$  as the root. This tree is traversed in a depth first manner and some pruning conditions are used to prune subtrees that do not have any motifs.

Algorithm qPMS7 of [10] extends the observation of qPMSPrune as follows: If  $M$  is any  $(\ell, d, q)$ -motif of the input strings  $s_1, \dots, s_n$ , then there exist  $1 \leq i \neq j \leq n$  and  $\ell$ -mer  $x \in s_i$  and  $\ell$ -mer  $y \in s_j$  such that  $M$  is in  $B_d(x) \cap B_d(y)$  and  $M$  is an  $(\ell, d, q-2)$ -motif of the input strings excluding  $s_i$  and  $s_j$ . qPMS7 considers every possible pair  $(i, j)$ ,  $1 \leq i, j \leq n$  and  $i \neq j$ . For a given pair  $(i, j)$ , every possible pair of  $\ell$ -mers  $(x, y)$  is considered (where  $x$  is from  $s_i$  and  $y$  is from  $s_j$ ). For a given  $x$  and  $y$ , the algorithm finds all the elements of  $B_d(x) \cap B_d(y)$  that are  $(\ell, d, q-2)$  motifs (with respect to input strings other than  $s_i$  and  $s_j$ ).  $B_d(x) \cap B_d(y)$  is explored by traversing an acyclic graph, denoted as  $\mathcal{G}_d(x, y)$ .  $\mathcal{G}_d(x, y)$  is traversed in a depth first manner. Here again effective pruning conditions are used to prune subgraphs of  $\mathcal{G}_d(x, y)$ .

For more details about the PMS algorithms, the readers are referred to the respective papers.

## An Experimental Validation of PMS Algorithms

Planted motif search is just one computational model for motifs. An important question is how efficient is this model in identifying motifs from real biological data. In fact the same question is relevant for any (computational or other) motif model. In [14], Tompa, et al. have evaluated the performance of 13 different motif finding programs: AlignACE, ANN-Spec, Consensus, GLAM, The Improbizer, MEME, MITRA, MotifSampler, Oligo/dyad-analysis, QuickScore, SeSiMCMC, Weeder and YMF. These programs were evaluated on several biological datasets (for which the motifs were known via experimental techniques) based on many different performance measures. Two of the performance measures employed were sensitivity and specificity. Sensitivity represents the fraction of sites that were correctly predicted and specificity represents the fraction of non-sites that were correct.

In [7], Sharma, et al. have evaluated the performance of PMS algorithms. In particular, they have employed the same 56 datasets that were used by Tompa, et al. [14]. As a result, Sharma, et al. have compared the PMS algorithms with the thirteen programs evaluated in [14]. Several versions of the PMS algorithms have been tested. One of these versions, namely, PMS SumMinD yields an average sensitivity of 28.8% and a specificity of 91.63% on all the 56 datasets. In comparison, the best of the 13 algorithms tested by Tompa, et al. [14], ANN-Spec, has an average sensitivity of 8.7% and a specificity of 98.22%.

## Our Motif Search Framework

In addition to the PMS algorithms, we deploy a motif search framework that uses the PMS algorithms as underlying routines. The motif search framework basically works as follows. The user inputs a set of sequences that contain motifs of interest. The framework runs a PMS algorithm (qPMS7 as of now) with different triples of the parameters  $(\ell, d, q)$  and collects all of the output motifs. These motifs are called candidate motifs. Then, it uses a score function that ranks the candidate motifs. The score function measures the significance of a candidate motif based on the probability that it occurs by random chance. Finally, the tool outputs the top 100 motifs with the highest scores. The score of a

candidate motif will be high if the probability that it occurs by random chance is low.

Since the run time of PMS algorithms is exponentially dependent on the parameter  $d$ , i.e. maximum number of mutations allowed, we let the user indirectly set the parameter through the computational preferences, “Quick Search” or “Full Search”. If the “Quick Search” option is chosen, then the parameter  $d$  is set to a ‘low’ value (3, specifically). Conversely if the “Full Search” option is chosen, then the parameter  $d$  is set to a higher value (7, specifically).

## Identifying Motif Instances in the Input Sequences

Once a motif is found, its instances in the input sequences will be located as follows. For each input sequence, the location of the motif instance in the input sequence is the place where the motif matches the most. The motif location can be done easily by scanning through the entire input sequence.

## Techniques to Identify Dyad Motifs

Eskin and Pevzner have presented an algorithm for finding dyads motifs [15]. This algorithm works as follows. Let the input sequences be  $s_1, s_2, \dots, s_n$  and let the length of each sequence be  $m$ . A dyad is characterized with the parameters  $(\ell_1, g_1, g_2, \ell_2, d, k)$ . Here  $\ell_1$  is the length of the first segment,  $\ell_2$  is the length of the second segment, the length of the gap between the two segments can be in the range  $[g_1, g_2]$ , and the dyad occurs in at least  $k$  out of the  $n$  sequences with a Hamming distance of at most  $d$ . For each input sequence  $s_i$ , the algorithm generates all the relevant  $\ell$ -mers (where  $\ell = \ell_1 + \ell_2$ ). Any such  $\ell$ -mer will be such that its prefix of length  $\ell_1$  will be an  $\ell_1$ -mer in some input sequence  $s + i$ , its suffix of length  $\ell_2$  will be an  $\ell_2$ -mer in the same sequence  $s_i$ , the prefix occurs to the left of the suffix, and the length of the gap between the prefix and the suffix is in the range  $[g_1, g_2]$ . Note that there are  $O(mn(g_2 - g_1))$  such  $\ell$ -mers. Let  $C$  be this collection of  $\ell$ -mers. After having generated these  $\ell$ -mers, they use the mismatch tree data structure to identify the  $\ell$ -mers that correspond to valid dyads. In particular, any  $\ell$ -mer will be output as a dyad if there is a  $d$ -neighbor of this  $\ell$ -mer that occurs in at least  $k$  of the input sequences.

We speed up the above algorithm exploiting the PMS1 algorithm. The improvement works as follows. We generate the  $\ell$ -mers for each sequence as in the algorithm of [15]. There are  $O(m(g_2 - g_1))$   $\ell$ -mers for each sequence. Let  $C_i$  be the collection of  $\ell$ -mers from sequence  $s_i$ , for  $1 \leq i \leq n$ . For each  $\ell$ -mer of  $C_i$  generate its  $d$ -neighborhood (i.e.,  $\ell$ -mers that are within a Hamming distance of  $d$  from the  $\ell$ -mer), for  $1 \leq i \leq n$ . Let  $C'_i$  be the collection of  $d$ -neighbors of all the  $\ell$ -mers of  $s_i$ , for  $1 \leq i \leq n$ . We can output  $d$ -neighbors that are in at least  $k$  of these collections. One way of finding such  $\ell$ -mers will be with the help of hashing. Another way is to make use of integer sorting. For example, we can sort each  $C'_i$  (for  $1 \leq i \leq n$ ), merge these sorted lists, and go through the merged list to count the number of sequences each such  $d$ -neighbor occurs in.

## Availability and Requirements

Project name: PMS - Panoptic Motif Search Tool. Project home page: <http://pms.engr.uconn.edu> or <http://motifsearch.com>. Licence: PMS tools will be readily available to any scientist wishing to use it for non-commercial purposes, without restrictions. The online tool is freely available without login.



## Author Contributions

Conceived and designed the experiments: HD SR. Performed the experiments: HD. Analyzed the data: HD SR. Contributed reagents/materials/analysis tools: HD SR. Wrote the paper: HD SR.

## References

- Jonassen I, Collins J, Higgins D (1995) Finding exible patterns in unaligned protein sequences. *Protein Science* 4: 1587–1595.
- Bailey TL, Boden M, Buske FA, Frith M, vGrant CE, et al. (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Research* 37: W202–W208.
- Neduva V, Russell R (2006) DILIMOT: discovery of linear motifs in proteins. *Nucleic Acids Research* 34: W350–W355.
- Davey NE, Shields DC, Edwards RJ (2006) SLiMDisc: short, linear motif discovery, correcting for common evolutionary descent. *Nucleic Acids Research* 34: 3546–3554.
- Edwards RJ, Davey NE, Shields DC (2007) SLiMFinder: A probabilistic method for identifying over-represented, convergently evolved, short linear motifs in proteins. *PLoS ONE* 2: e967.
- Lieber DS, Elemento O, Tavazoie S (2010) Large-scale discovery and characterization of protein regulatory motifs in eukaryotes. *PLoS ONE* 5: e14444.
- Sharma D, Rajasekaran S, Dinh H (2011) An experimental comparison of PMSPrune and other algorithms for motif search. *CoRR* abs/1108.5217.
- Rajasekaran S (2009) Computational techniques for motif search. *Frontiers in Bioscience* 14: 5052–5065.
- Pisanti N, Carvalho AM, Marsan L, Sagot MF (2006) RISOTTO: Fast extraction of motifs with mismatches. *Proceedings of the 7th Latin American Theoretical Informatics Symposium*: 757–768.
- Dinh H, Rajasekaran S, Davila J (2012) qPMS7: A fast Algorithm for finding (l; d)-motifs in DNA and protein sequences. *PLoS ONE*, 7(7): e41425.
- Dinh H, Rajasekaran S, Kundeti V (2011) PMS5: an efficient exact algorithm for the (l; d)-motif finding problem. *BMC Bioinformatics* 12(410).
- Davila J, Balla S, Rajasekaran S (2007) Fast and practical algorithms for planted (l; d) motif search. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*: 544–552.
- Rajasekaran S, Balla S, Huang CH (2005) Exact algorithms for planted motif challenge problems. *Journal of Computational Biology* 12(8): 1117–1128.
- Tomba M, Li N, Bailey TL, Church GM, Moor BD, et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology* 23(1): 137–144.
- Eskin E, Pevzner P (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics* 18: 354–363.