A Graph Clustering Approach to Weak Motif Recognition

Christina Boucher, Daniel G. Brown, and Paul Church

D.R. Cheriton School of Computer Science
University of Waterloo
{caboucher,browndg,pchurch}@cs.uwaterloo.ca

Abstract. The aim of the *motif recognition problem* is to detect a set of mutually similar subsequences in a collection of biological sequences. *Weak motif recognition* is where the sequences are highly degenerate. Our new approach to this problem uses a weighted graph model and a heuristic that determines high weight subgraphs in polynomial time. Our experimental tests show impressive accuracy and efficiency. We give results that demonstrate a theoretical dichotomy between cliques in our graph that represent actual motifs and those that do not.

1 Introduction

Understanding the structure and function of genomic data remains an important biological and computational challenge. *Motifs* are short subsequences of genomic DNA responsible for controlling biological processes, such as gene expression. Motifs with the same function may not entirely match, due to mutation. The *motif consensus* of the instances is a sequence representing the shared pattern. Given a number of DNA sequences, *motif recognition* is the task of discovering motif instances in sequences without knowing their positions or pattern. This problem becomes increasingly difficult as the number of allowed mutations grow. Weak motif recognition addresses the difficult case when many degenerate positions are allowed. Many useful versions of motif recognition are NP-complete, and therefore are unlikely to have polynomial-time algorithms.

Pevzner and Sze define the weak motif recognition problem concretely, illustrating the limitations of motif recognition programs. In 2000, most methods were capable of finding motifs of length 6 with no degeneration but failed to detect motif instances of length 15 with 4 degenerate positions in a random sample containing 20 sequences of length 600 [8]. Since this "challenge problem" was defined, many approaches have been developed to detect motifs with a relatively large number of degenerate positions.

We describe a new approach for this problem, and provide theoretical and experimental results that support our novel motif recognition algorithm. Our algorithm, MCL-WMR, builds an edge-weighted graph model of the given motif recognition problem and uses a graph clustering algorithm to quickly determine important subgraphs are to be searched for valid motifs. Synthetic data has

R. Giancarlo and S. Hannenhalli (Eds.): WABI 2007, LNBI 4645, pp. 149–160, 2007.

shown that MCL-WMR has competitive running time capabilities and accuracy. An added advantage of MCL-WMR is the ability to detect multiple motif instances.

The efficiency of MCL-WMR lies in the use of the *Markov Cluster algorithm (MCL)* to quickly find dense subgraphs likely to contain a motif. These subproblems are then solved optimality via dynamic programming. Extracting important subgraphs is in the spirit of WINNOWER, the combinatorial algorithm created by Pevzner and Sze [8], which builds a similar graph model and eliminates spurious edges sequentially. Our algorithm eliminates complete subgraphs and hence, avoids considering edges individually.

One of the main contributions of the creation of MCL-WMR is the introduction of a novel model for motif recognition. Previous algorithms and programs search exhaustively or probabilistically on an unweighted graph or string. Due to the lack of information contained in these models, the required search requires extensively computation. By considering a weighted graph model, we narrow the search dramatically to easy problems. We argue that there exists a dichotomy between the weight of cliques corresponding to actual motifs and that of cliques which do not, and suggest this separation can be used to filter data to be searched.

2 Previous Approaches to Weak Motif Recognition

The limitations of the existing motif recognition programs were first highlighted by Pevzner and Sze, who identified "challenge" problems in motif discovery [8]. We approach the problem from a similar combinatorial perspective and hence, consider the following combinatorial formulation.

Definition 1. *The (l, d)-motif problem: Let $S = \{S_1, \ldots, S_m\}$ be a set of n length DNA sequences, and M be the motif consensus, a fixed and unknown sequence of length l. Suppose that M is contained in each S_i, corrupted with at most d substitutions, so their Hamming distance is at most d. The aim is to determine M and the location of the motif instance in each sequence.*

The Hamming distance between two sequences s_i and s_j is $H(s_i, s_j)$. The *weak motif recognition problem* is to find the motif instances when the number of degenerate positions d is large in relation to the motif length l; well-known weak motif recognition problems exist when the motif instances are $(9, 2)$ $(11, 3)$, $(15, 4)$, and $(18, 6)$, with 20 DNA random sequences, each 600 nucleotides long. Although the strength of the motif leads to an increased or decreased inherent difficulty, varying the background sequence length is also important. As the number of sequences increases, the number of noisy l-mers increases: detection of the motif instances becomes increasingly difficult, and spurious motifs are more likely to occur. Also, as the sequence length grows, the number of near-motifs will also increase dramatically.

Existing software programs developed for motif finding use either a heuristic or enumeration approach. Heuristic methods attempt to maximize a score function representative of how likely a particular subsequence is a motif instance;

they are often unsatisfactory for weak motifs because they get trapped in local maxima. Pevzner and Sze developed WINNOWER and SP-STAR for weak motif recognition. WINNOWER creates a graph representation with a vertex for every occurring l-mer and an edge between all pairs of vertices that are at most $2d$ distance apart; spurious edges are deleted to reveal sets of vertices whose corresponding subsequences are possible motif instances [8]. Due to spurious edges, the running time is prohibitively large and grows immensely as motif strength weakens or subsequence length or number increases [5].

Sze *et al.* [10] extend upon the graph formation of WINNOWER [8]: they formulate motif finding as finding cliques in k-partite graphs, with the additional requirement of a string s close to every motif instance. They hypothesize that this approach provides a better formulation to model motifs than using cliques alone; the use of k-partite graphs lends itself to be solved exactly and efficiently by a divide-and-conquer algorithm. Experimental results demonstrate that the approach is feasible on difficult motif finding problems of moderate size [10].

Buhler and Tompa [1] developed a heuristic algorithm called PROJECTION that projects every occurring l-mer onto a smaller space by hashing. The hash function is based on k of the l positions that are selected at random when the algorithm begins. After the initialization step, a consensus is derived for each grouping of l-mers, and expectation maximization is used for refinement. PROJECTION does significantly better than other program but its accuracy is dependent on an user-defined input parameter [1]. As m becomes larger, PROJECTION recovers motif instances slower as m increases; hence, the running time and accuracy of is very sensitive to changes in m.

An obvious method to detect motif instances of length l is to enumerate all 4^l possible motif consensus sequences, count occurrences, and calculate a significance value for each of the considered l-mers or count instances of them or see if they satisfy a requirement as in the (l, d) problem. These algorithms are guaranteed to find the best motif (or the most probable one, in the case of maximizing a likelihood function), but their running times become prohibitively slow for large degenerate motifs. To tackle more significant motif recognition problems, enumeration methods have been created that consider only oligomers which are present in the given data sets.

SP-STAR, developed by Pevzner and Sze [8], does an enumerative search but only over the occurring data rather than the entire space of 4^l l-mers.; however, we note that the number of sequences to be searched is approximately $\binom{l}{d}3^d$. SP-STAR was successful in finding (15,4)-motif instances in data sets containing 20 DNA sequences, each of which has maximum length 700 but failed to have reasonable accuracy when the sequence length exceeded 700 [8].

3 System and Methods

MCL-WMR involves three stages: graph construction, clique finding using graph clustering, and recovering the motif instances and their consensus. The construction of MCL-WMR is as follows: a reference sequence S_r is chosen randomly

from the data set and for each l-length subsequence of S_r a graph G_r is built from comparing that subsequence with all other possible l-length subsequences in the data set $S_1, \ldots, S_{r-1}, S_{r+1}, \ldots, S_m$. The entire graph G is the union of these subgraphs $G_1, \ldots, G_{m-l}$. We use MCL to generate subgraphs which contain vertices that are highly inter-related. From these clusters of vertices, we generate the positions of the possible motif instances and their corresponding motif consensus. The algorithm terminates when a motif is found. In order to increase the probability a motif is found, we minimize searching subgraphs with low probability of containing a motif; hence, the adjacency subgraphs are not clustered and searched in a sequential manner.

3.1 Graph Construction

In our graphical representation of the data set, each subsequence of length l is represented by a vertex and the construction of our graph ensures that the motif instances represented by vertices in the graph are connected to each other and form a clique of size m (though the converse need not hold). The vertex set contains a vertex $v_{i,j}$ representing the l-length subsequence in sequence i starting at position j, for each i and $j = 1, 2, \ldots, n - l + 1$. Each pair of vertices $v_{i,j}$ and $v_{i',j'}$, for $i \neq i'$ is joined by an edge when the Hamming distance between the two represented subsequences is at most $2d$. An edge between vertices at distance k has weight $l - k$ for $d < k \leq 2d$, or $10(l - k)$ for $k \leq d$. This emphasizes subsequences at small distances. This graph is represented by a symmetric adjacency matrix, constructed in $O(m^2(n - l)(n + l))$ time. The graph is m-partite so a clique of size m contains exactly one vertex from each sequence. We reduce the size of the instance being passed to MCL by considering subgraphs $\{G_0, G_1, \ldots, G_{m-l}\}$, where G_i is the subgraph induced by a reference vertex, denoted as $v_{R,i}$, and its neighbors (for some arbitrary choice of reference sequence R) instead of searching all of G at once.

3.2 Using Clustering to Find Motifs

A *clustering* of graph $G(V, E)$ is a decomposition of V into subsets of highly intra-connected vertices. A *good clustering* of a graph is an approximation of a partitioning of the graph into cliques. A clique corresponding to a motif will exist in one of the subgraphs of G since each motif instance appears as a vertex in a clique of size m. We use MCL [11] to cluster the sets of vertices to determine subgraphs that are highly intra-connected with high-weight edges, and scarcely inter-connected and thus, likely to correspond to a motif instance. MCL can handle larged, undirected weighted graphs efficiently. The idea underlying the MCL algorithm is that dense subgraphs correspond to regions where the number of k-length paths is relatively large, for small k. Random walks of length k have higher probability for paths beginning and ending in the same dense region than for other paths.

3.3 Recovering Motifs

MCL identifies dense high-scoring regions of the subgraph G_i; we filter the subgraphs obtained from MCL to subgraphs that have high probability of containing

a motif. A clique in G that represents a motif instance must have size n and weight greater than or equal to $(l-2d)\binom{m}{2}$ since each pair of vertices are adjacent. We filter out clusters that do not meet these criteria. Clusters that pass this test may contain multiple cliques formed by choosing different subsets of n cluster vertices, or possibly no cliques at all. We identify all ways of forming a clique from the cluster vertices by using the m-partite nature of the graph to explore all possible cliques with a depth-first search. As the number of cliques can be exponential in the cluster size in the worst case, this step becomes a bottleneck for problem sizes such as $(18, 6)$, where MCL returns large clusters.

For each clique, we test if it represents a motif instance by attempting to build a motif consensus that has distance at most d to every vertex. We do this by building up a list of possible consensuses and the number of mismatches to each vertex for each possibility, one character at a time. Once a candidate consensus has $d+1$ mismatches to some vertex, it is discarded. Although the space of 4^l possible consensus strings is very large, in practice the list is pruned very rapidly on the $d+1$st character, i. e. after reaching size 4^d.

4 Analysis of Graph-Theoretic Model

To validate our weighted graph approach, we show the existence of a separation between the total weight of a clique corresponding to a motif and that of a clique that does not. We demonstrate theoretically that the total weight of a clique corresponding deviates from the mean with low probability. Empirical results support this, and also show that there exists some separation between cliques that can be extended to motifs and those that cannot.

4.1 Analysis of the Weight of a Clique Containing a Motif

Consider a clique C containing a motif. Define the *weight* of an edge to be l minus the Hamming distance between the sequences corresponding to the endpoints of the edge. Let W be the random variable for the sum of each of the $\binom{m}{2}$ edge weights in C. Without loss of generality, let $v_1, v_2, \ldots v_m$ be the set of m vertices in C corresponding to sequences $s_1, \ldots, s_m$. We seek $E[W]$ and a tail bound a large deviations from the mean. Let W_i be the expected value of the random variable W given that the first i subsequences in C are known.

Theorem 1. *The expected weight of a clique in G, which models a random (l, d)-motif recognition problem containing m sequences, is*

$$E[W] = \binom{m}{2}\left(l - \frac{1}{\beta^2}\sum_{a=0}^{d}\sum_{b=0}^{d}\binom{l}{b}\binom{l}{a}3^{a+b}\left(a+b-\frac{4ba}{3l}\right)\right)$$

where $\beta = \sum_{i=0}^{d}\binom{l}{i}3^i$.

Proof. Given an (l, d) motif, we aim to compute the expected value of the clique's total weight, $E[\sum_{i=1}^{m}\sum_{j=i+1}^{m}(l - H(s_i, s_j))]$. Let μ_e be $E[H(s_i, s_j)]$ for any pair

of sequences s_i and s_j, where (v_i, v_j) is an edge in a clique that contains a motif and s_i and s_j are unknown.

$$E[W] = E\left[\sum_{\forall v_i, v_j, i<j} (l - H(s_i, s_j))\right] = \binom{m}{2}\mu_e$$

We choose the m sequences uniformly from the $\beta = \sum_{i=0}^{d}\binom{l}{i}3^i$ possible choices. Let α_i denote the Hamming distance between vertex v_i and the consensus S. The expected weight of an edge depends on the distance of the two subsequences from the consensus, so we break the expectation into pieces:

$$\begin{aligned}\mu_e &= \textstyle\sum_{\alpha_i=0}^{d} Pr[H(S,s_i)=\alpha_i]E[H(s_i,s_j)|H(S,s_i)=\alpha_i] \\ &= \textstyle\sum_{\alpha_i=0}^{d} Pr[H(S,s_i)=\alpha_i]\cdot\sum_{\alpha_j=0}^{d} Pr[H(S,s_j)=\alpha_j]|| \\ &\quad E[H(s_i,s_j)|H(S,s_i)=\alpha_i, H(S,s_j)=\alpha_j] \\ &= \textstyle\sum_{\alpha_i=0}^{d} \frac{\binom{l}{\alpha_i}3^{\alpha_i}}{\beta}\sum_{\alpha_j=0}^{d}\frac{\binom{l}{\alpha_j}3^{\alpha_j}}{\beta}E[H(s_i,s_j)|H(S,s_i)=\alpha_i, H(S,s_j)=\alpha_j]\end{aligned}$$

The remaining problem is to compute the expected Hamming distance between s_i and s_j, knowing that the strings consist of copies of S with a and b positions mutated, respectively. If a position was mutated in neither string it is a match; if a position was mutated in one string but not the other it is a mismatch; if a position was mutated in both strings, it is a match with probability $\frac{1}{3}$.

If s_i is fixed, s_j consists of b mutations that each either hit one of the a mutated positions in s_i or one of the other $l-a$ positions, sampled without replacement. The number that hit the a mutated positions in s_i follows a hypergeometric distribution with mean $\frac{ba}{l}$. If the number of hits to mutated positions is c, the expected total number of mismatches is: $b-c$ positions that hit among the $l-a$ non-mutated positions in s_i, $a-c$ positions among the a mutated positions of s_i that were *not* hit, and $\frac{2}{3}c$ mismatches from the hits among the mutated positions, for a total of $(b-c)+(a-c)+\frac{2}{3}c = a+b-\frac{4}{3}c$. Therefore, $E(H(s_i,s_j)|H(S,s_i)=a, H(S,s_j)=b) = a+b-\frac{4ba}{3l}$. Let μ_e be $l-\frac{1}{\alpha^2}\sum_{a=0}^{d}\binom{l}{a}3^a\sum_{b=0}^{d}\binom{l}{b}3^b\left(a+b-\frac{4ba}{3l}\right)$, the expected weight of a single edge. We have the following:

$$\begin{aligned}E[W] &= \textstyle\binom{m}{2}\left(l-\frac{1}{\alpha^2}\sum_{\alpha_i=0}^{d}\binom{l}{\alpha_i}3^{\alpha_i}\sum_{\alpha_j=0}^{d}\binom{l}{\alpha_j}3^{\alpha_j}\left(\alpha_i+\alpha_j-\frac{4\alpha_i\alpha_j}{3l}\right)\right) \\ &= \textstyle\binom{m}{2}\left(l-\frac{1}{\alpha^2}\sum_{\alpha_i=0}^{d}\sum_{\alpha_j=0}^{d}\binom{l}{\alpha_j}\binom{l}{\alpha_i}3^{\alpha_i+\alpha_j}\left(\alpha_i+\alpha_j-\frac{4\alpha_i\alpha_j}{3l}\right)\right)\end{aligned}$$

We are able to easily bound the variance of W by first demonstrating that $W_0, W_1, \ldots, W_m$ is a martingale sequence and next, applying Azuma's inequality to determine the probability of a specific deviation.

Theorem 2. *Consider the (l,d)-motif recognition problem containing m sequences. Let W be the sum of the $\binom{m}{2}$ edge weights in an arbitrary clique in G_{motif} that contains a motif and let μ_W be the expected weight of a clique in G that contains a motif, then for any $\lambda > 0$,*

$$Pr[|W-\mu_W| \geq \lambda] \leq 2\exp\left(-\frac{\lambda^2}{2d^2(m+1)}\right)$$

Proof. The mean of W has been previously defined, here we concentrate on proving the tail bound. Recall that W_i is the expected value of the random variable W given that the first i subsequences in C are known and hence, the distances between the consensus S and the first i vertices are known. Without loss of generality we choose a consensus S and let F_i be the σ-field generated by the random choice of the subsequence i from the set of all subsequences at most distance d from the consensus and hence, α_i randomly chosen with the same probability. It follows that $W_i = [W|\mathrm{F}_i]$, since W_i denotes the conditional expectation of W knowing the first i subsequences. Therefore, $W_0, W_1, \ldots, W_m$ is a martingale sequence [6], with $W_0 = E[W]$ and $W_m = W$.

We now focus on the value $W_i - W_{i-1}$. Let $\Delta_{i,e}$ be the change in the random variable representing the weight of an edge e from knowing the first $i-1$ sequences to knowing the first i sequences. The value of $\Delta_{e,i}$ is non-zero for edges where the sequence corresponding to one of the endpoints of that edge was previously not known and is now known. Each vertex in the clique is adjacent to $m-1$ vertices. For $i-1$ the corresponding sequences were known and for $n-i$, the corresponding sequences were unknown. All other $\binom{m}{2} - m + 1$ remaining edges in the clique have no change. The expectation of the weight of the edge can change by at most d.

$$|W_i - W_{i-1}| \leq d(m-1)$$

The random variables $W_0, W_1, \ldots, W_m$ form a martingale with $W_0 = E[W]$ and $W_m = W$ and that $|W_i - W_{i-1}| \leq d(m-1)$ Therefore, we can invoke Azuma's inequality to give us the following for any $\lambda > 0$:

$$Pr\left[|W - \mu_W| \geq \lambda\right] \leq 2\exp\left(-\frac{\lambda^2}{2\sum_{i=0}^{m} d^2} =\right) = 2\exp\left(-\frac{\lambda^2}{2d^2(m+1)}\right)$$

We compare the theoretical tail bound with the distribution of values obtained from MCL-WMR; Figure 4.1 demonstrates that the distribution of the values of W approach the normal distribution in the limit with the mean value centered at 897. This corresponds to the theoretical mean of 900.1 calculated using the result from Theorem 1. The weight of cliques that do not represent valid motifs appears to follow a normal distribution but their mean weight is slightly lower, approximately 885. This result, shown in Figure 4.1, was determined by generating the weight of cliques that do not correspond to motifs in 100 random data sets but these were discovered using MCL-WMR and so are likely are not a uniform sample of such cliques.

These results demonstrate a partial separation between the weight of cliques representing motifs and those that do not, which can be exploited to efficiently find dense subgraphs that are of interest. As highlighted in Figure 4.1, we use the weight to determine which subgraphs a further search for valid motifs is necessary. Further, Figure 4.1 demonstrates that as the value of m increases this separation will become more apparent since the deviation of the weight of cliques corresponding to motifs will occur with less probability, and the weight of the cliques will

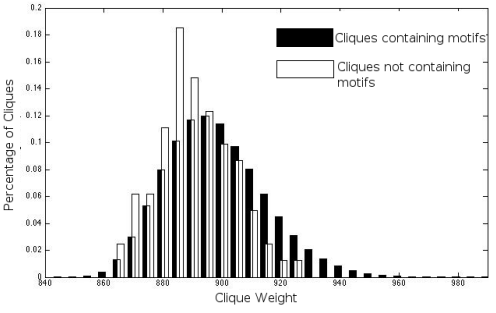

Fig. 1. Distribution of the weight of cliques containing a motif consensus and the distribution of the weight of cliques not containing a motif consensus. The data for non-motif cliques was generated was generated by running MCL-WMR 100 times, calculating the total weight of the clique, and generating a histogram of these values. The data is given for the (15,4) motif problem instance with $m = 15$.

become more centralized around the mean. Similar experimental tests were completed to demonstrate the relationship between the weight of spurious cliques when $m = 15$ and when $m = 50$, specifically, we ran MCL-WMR 100 times with $n = 800$, $l = 15$, and $d = 4$ and determined cliques that did not correspond to valid motifs. We found no spurious cliques in the data sets when $m = 50$, agreeing with our intuition that very few spurious cliques occur randomly in the data set when m becomes large. We should further note our confidence in MCL-WMR being able to detect cliques–both spurious and those corresponding to motifs–this is due to the accuracy is detecting the embedded motifs (see Section 4.2 details concerning these experimental tests). These results also suggest that when m is relatively large we can be more certain than any cliques found correspond to valid motifs; an attribute that should be further explored.

4.2 Discussion of Complexity

A few interesting observations can be made regarding the complexity of the algorithm and the quality of its solutions. Finding cliques of maximum size in a given input graph is NP-complete and thus, unlikely to be solved in polynomial-time [3]. Further, the results from Chen *et al.* [2] show that unless unlikely consequences occur in parameterized complexity theory, the problem of finding maximum–size cliques cannot be solved in $n^{o(k)}$ time. Thus, finding cliques of a specific size k is not likely to be computationally feasible for graphs of significant size. The best known algorithm for finding cliques of size k in a given input graph runs in time $O(m^{ck/3})$, where c is the exponent on the time bound for multiplying two integer $m \times m$ matrices; the best known value for is c is 2.38 [7]. The runtime for the straightforward algorithm that checks all size k subsets is $O(m^{k+2})$ and is the one to be most likely to be implemented in practice. The runtime of the algorithm of Yang and Rajapakse [12], a dynamic programming clique finding algorithm, is $O(m(nA^2 + A^{n-1}p^{2n-5})$, where $A = m\sum_{i=0}^{2d}\binom{l}{i}(3/4)^i(1/4)^{l-i}$, m is the length of each sequence and n is the number of sequences. This runtime

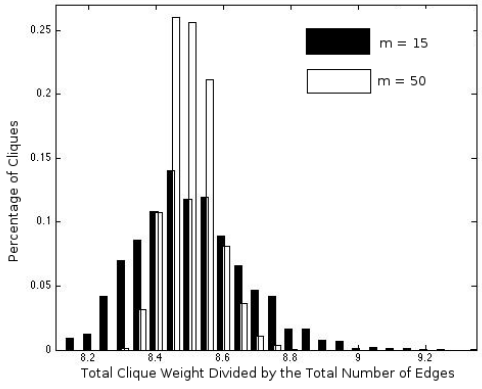

Fig. 2. Distribution of average edge weights in cliques corresponding to actual motifs of size 15 and 50. The data is given for the $(15, 4)$ motif problem with $n = 600$.

reflects the steep computational expense required to find cliques of a given size for an input graph. Similarly, the estimated runtime of the WINNOWER algorithm is $O((mD)^4)$, where D is approximately 30 for the challenge problem [8].

The time required by MCL-WMR to find a solution is not affected by the length of the motif that is to be discovered, whereas this is true for many other methods. Rather, it is the weakness of the motif–that is, the probability of the pairwise motif similarity occurring randomly–that has the most impact on the complexity of the algorithm. The increased probability of a clique of high weight affects the runtime of MCL-WMR since the exponential-time algorithm required to determine in a high cluster or subgraph contains a motif instance. We can compare the computational complexity of these programs by considering the required runtime of MCL-WMR of the three sequential steps–that is, the computational time required to construct the graph, finds all cliques of size m, and determine the motifs and consensus.

MCL-WMR uses the MCL algorithm that runs in time $O(N^3)$ where N is the number of vertices in the input graph [11] to find dense subgraphs. Hence, the most computationally expensive step of MCL-WMR is the clique-finding algorithm that serves the dense subgraphs for cliques corresponding to valid motifs and increases in computation time with the number of vertices. Other graph-based methods for motif finding rely on enumeration methods to find dense subgraphs; for example, WINNOWER requires each edge to be checked and the algorithm of Yang and Rajapakse uses dynamic programming on the complete graph,

5 Experimental Results

We tested MCL-WMR on synthetic problem instances generated according to the embedded (l, d)-motif model. We produce problem instances as follows: first we choose a random motif consensus of length l, and pick m occurrences of the motif

Table 1. Comparison of the performance on a range of (l, d)-motif problems with synthetic data, where $n = 600$ and $m = 20$. The average performance of MCL-WMR on the eight different problem instances, generated as specified, are given. Data for WINNOWER and SP-STAR is the average of eight random instances given by Pevzner and Sze [8], while PROJECTION is the average of 100 random problem instances where the projection size is 7 and the bucket size is 4 give by Bulher and Tompa [1].

l	d	PROJECTION	SP-STAR	WINNOWER	MCL-WMR	Time
10	2	0.80	0.56	0.78	1.00	54 ± 10.8
11	2	0.94	0.84	0.90	1.00	30 ± 10.6
12	3	0.77	0.33	0.78	1.00	205 ± 11.0
13	3	0.94	0.92	0.92	1.00	65 ± 10.4
14	4	0.71	0.02	0.02	1.00	806 ± 71.3
15	4	0.93	0.73	0.92	1.00	220 ± 17.2
17	5	0.93	0.69	0.03	1.00	704 ± 67.2
18	6	N. A.	N. A.	N. A.	1.00	20605 ± 534.3

by randomly choosing d positions per occurrence and randomly mutating the base at each. Lastly, we construct m background sequences of length n and insert the generated motifs into a random position in the sequence. For each of the (l, d) combinations, 100 randomly generated sets of input sequences ($n = 600$ and $m = 20$) were generated. This generation corresponds to the "FM" model used in the challenge problem by Pevzner and Sze and the results concerning PROJECTION by Buhler and Tompa. All empirical results were obtained on a desktop computer with a 2.0 GHz AMD Athlon 64 bit processor with 512 KB cache and 1 GB RAM, running Debian Linux. The time is the number of CPU seconds.

One of the main advantages of MCL-WMR is the accuracy of the results even for hard problems. A metric, referred to as *performance coefficient*, is used to gauge the accuracy of the algorithm and is defined as $\frac{K \cap P}{K \cup P}$, where K is the set of ls nucleotides in motif instances and P is the set of ls nucleotides in the proposed motif instances. A performance coefficient of 0.75 or greater is acceptable for algorithms not guaranteeing exact accuracy; improved algorithms return results with coefficients between 0.9 and 0.95.

Table 1 compares the performance of MCL-WMR with that of previous motif finding programs on sets of eight random problem instances. We give the average performance coefficient for MCL-WMR and the competing programs, the mean runtime, and the range of runtimes for each set of motif problem instances. For comparison, we give the performance coefficients for WINNOWER, SP-STAR and PROJECTION. The data for these corresponding algorithms was collected by Pevzner and Sze [8] and Bulher and Tompa [1]. Our program found the exact location of a motif instance every single time and hence, the coefficient is 1; other programs typically were only approximate in discovering the motifs. The computation time of previous programs that find the exact solution becomes unacceptable as the motifs become degraded beyond the $(15, 4)$ problem [9]. The main advantage to our tool is the time required to solve the extremely difficult challenge problems–that is $(17, 5)$ ans $(18, 6)$ problem–is substantially better to the running time of previous algorithms.

Table 2. Comparison of the time required to solve the $(15, 4)$-motif problem with 20 sequences of varying length, of MCL-WMR and PROJECTION; n denotes the sequence length, which varies from 600 to 2000. The running times are obtained by averaging the time to obtain a solution on 8 different instances of the problem. Data for PROJECTION was collected from King *et al.* [4].

n	PROJECTION	MCL-WMR
600	6.6 ± 1.0	50 ± 17.6
800	27 ± 4	118 ± 39.9
1000	82 ± 25	228 ± 67.4
1200	250 ± 60.0	407 ± 78.8
1400	600.6 ± 140.0	706 ± 138.6
1600	1000 ± 200	1043.4 ± 80.51
1800	1435 ± 353.0	1652 ± 342.7
2000	1891 ± 600.0	2078 ± 432.2

The performance coefficient of MCL-WMR is greater than that of the previous algorithms in every line of Table 1. MCL-WMR correctly solved planted $(11, 2)$, $(13, 3)$, $(15, 4)$, $(17, 5)$ and $(18, 6)$ on all data sets–in these cases, the planted motif and motif occurrences at least as strong as planted motifs. WINNOWER, PROJECTION, and SP-STAR achieve acceptable performance on the $(11, 2)$, $(13, 3)$ and $(15, 4)$ problem instances when the sequence length is less than or equal to 600 and the number of sequences is less than or equal to 20, however, all fail on the $(18, 6)$ and $(19, 6)$ problem, and WINNOWER and SP-STAR fail on the $(16, 5)$ and $(17, 5)$ problem instances. The performance of MCL-WMR is most eminent on the more difficult planted $(14, 4)$, $(16, 5)$, $(17, 5)$ and $(18, 6)$ motif problems when compared to results from previous algorithms. WINNOWER and SP-STAR typically failed to find the planted motifs and PROJECTION often failed to have acceptable performance on the more difficult cases of the challenge problem [1] and hence, MCL-WMR's performance substantially exceeded that of previously algorithms.

We evaluated the performance of MCL-WMR on problem instances with longer background sequences–that is, problems where n varies from values greater than 600. As the length of the sequences increase, the number of randomly occurring l-mers increases; specifically, the increase in n, increases the probability of cliques of high-weight occurring. Due to the increase in noise and hence, difficulty in detecting true motifs, MCL-WMR will recover motifs more slowly. Our results are comparable to the results of PROJECTION, as can be seen in Table 2, MCL-WMR maintains its speed advantages as n increases. Considering the $(15, 4)$ problem and fixing the number of sequences to be 20, the performance of WINNOWER breaks at length 700, and SP-STAR breaks when the length is 800 to 900. Table 2 demonstrates that MCL-WMR has comparable running time to PROJECTION for lengths above 1400, and in any case higher. For smallest lengths PROJECTION appears to be faster. We should further note that MCL-WMR achieves a performance ratio of 1.0 whereas, PROJECTION achieved a performance ratio around 0.93.

6 Conclusion

We propose an efficient algorithm for motif recognition with the specific purpose of solving more difficult problems when the motif signal is weak due to a large amount of degeneration. We demonstrate promising results on synthetic data. Specifically, we showed promising running time and accuracy for all challenge problems, with most-impressive improvement on the $(14, 4)$, $(17, 5)$ and $(18, 6)$ problems. Previous algorithms lack accuracy, the ability for the running time to scale with the length and number of sequences, and achieving a reasonable running time for all challenge problems.

We have shown that a novel model for motif recognition can dramatically influence the algorithmic ability and efficiency. By changing the graphical model to incorporate edge weights, we exploit theoretical results demonstrating the existence of a separation between the weights of cliques corresponding to valid motifs and the weights of those that do not, and obtain improved search techniques. Our theoretical work and empirical data show a large percentage of the cliques corresponding to valid motifs have total weight in a narrow range. This helps us distinguish cliques containing valid motifs and spurious cliques. We expect interesting theoretical results lie within study of this weighted graph model, along with further exploitation of theoretical results for the problem.

References

1. Buhler, J., Tompa, M.: Finding motifs using random projections. J. Comput. Biol. 9(3), 225–242 (2002)
2. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Linear FPT reductions and computational lower bounds. In: Proc. Sym. on Theory of Comp., pp. 212–221 (2004)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., New York, NY (1979)
4. King, J., Cheuny, W., Hoos, H.H.: Neighbourhood Thresholding for Projection-Based Motif Discovery. Bioinfo. (to appear)
5. Liang, S., Samanta, M.P., Biegel, B.A.: cWINNOWER algorithm for finding fuzzy DNA motifs. J. Bioinfo. Comput. Biol. 2(1), 47–60 (2004)
6. Motwani, R., Raghavan, R.: Randomized Algorithms. Cambridge University Press, New York, NY (1995)
7. Niedermeier, R.: Invitation to fixed-parameter algorithms. Habilitation thesis, Universität Tübingen (2002)
8. Pevzner, P., Sze, S.: Combinatorial approaches to finding subtle signals in DNA sequences. Proc. Int. Conf. Intell. Syst. Mol. Biol (ISMB00) 8, 344–354 (2000)
9. Styczynski, M.P., Jensen, K.L.: An extension and novel solution to the $(1, d)$-motif challenge problem. Gen. Info. 15(2), 63–71 (2004)
10. Sze, S., Lu, S., Chen, J.: Integrating sample-driven and patter-driven approaches in motif finding. In: Jonassen, I., Kim, J. (eds.) WABI 2004. LNCS (LNBI), vol. 3240, pp. 438–449. Springer, Heidelberg (2004)
11. van Dongen, S.: Graph clustering by flow simulation. PhD thesis, University of Utrecht (May 2000)
12. Yang, X., Rajapakse, J.: Graphical approach to weak motif recognition. Gen. Info. 15(2), 52–62 (2004)