FURTHER RESULTS ON MESSAGE-PASSING ALGORITHMS FOR MOTIF FINDING

Sang Hyun Lee, Haris Vikalo, and Sriram Vishwanath

Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA
Email: {shlee, hvikalo, sriram}@ece.utexas.edu

ABSTRACT

A new class of message-passing algorithms for motif finding is presented. Motif finding is the problem of identifying a collection of common subsequences within a given set of DNA sequences. It can be cast as an integer linear program (ILP). Message-passing techniques are a computationally efficient alternative to the often infeasible combinatorial solutions to the ILP. We introduce a new graphical representation of the ILP formulation of the problem, and use it to develop new message-passing algorithms for motif finding. Simulation results demonstrate that the new algorithms have better performance and convergence properties than the previously proposed solutions.

Index Terms: message passing, motif finding

1. INTRODUCTION

Reverse-engineering of gene regulatory networks is one of the most important challenges in computational biology. Gene regulatory networks (GRNs) are systems of biomolecular components (genes, mRNA, proteins) that interact with each other and through those interactions determine gene expression levels - i.e., determine the rate of gene transcription to mRNA and, consequently, the rate of mRNA translation to proteins. Motifs, short DNA segments adjacent to a regulated gene, play an important role in the cellular regulatory mechanism. They bind transcription factors and, in doing so, enable RNA Polymerase to transcribe genes to mRNAs. Identifying motifs is an important step towards understanding GRNs.

On a related note, recent development of DNA and protein microarrays enabled time-series studies of gene expression levels. In particular, microarrays may provide temporal samples of the abundance of mRNAs transcribed from a selected set of genes. Now, if the expression levels of several genes (i.e., the amounts of their corresponding mRNAs) are correlated, this is often an indication of the co-existence of the same motif adjacent to each one of the genes. So, given a set of p DNA sequences suspected to contain a common motif, the problem is to identify a subsequence within each sequence such that a similarity measure among the subsequences is maximized. Typically, this boils down to minimizing the sum of pairwise distances between pairs of said subsequences. This can easily be rephrased as an integer linear programming problem, and formulated as an optimization on a weighted p-partite graph [1]. In particular, the objective of the problem reduces to choosing a node from each partition so that the weight of the induced subgraph is minimized. However, finding the exact solution to an integer linear program is computationally rather demanding.

On another note, message-passing algorithms have recently attracted a lot of interest. These are distributed and decentralized techniques which offer a significant computational advantage over the traditional centralized combinatorial solution methods. Typically, they are designed to efficiently compute conditional marginal probabilities (or determine the assignment with the highest probability) of a joint discrete probability distribution defined on a graphical model of the constraint satisfaction problem. In [2], we reported our preliminary studies of message-passing for motif finding. In this paper, we develop new and improved algorithms with better performance and convergence properties.

2. MODEL AND PROBLEM FORMULATION

We assume that a motif is an $l$-bases long subsequence of a DNA sequence. Moreover, assume that we are given $p$ DNA sequences, each of length $N_0$. The goal of the motif finding problem is to find a subsequence of length $l$ within each DNA sequence such that a similarity measure among the subsequences is maximized. Typically, this translates to the requirement that the sum of the pairwise distances between any pair of subsequences of length $l$ be minimized.

We will find it useful to rephrase the motif finding problem using a graphical model. Define a weighted $p$-partite graph $G = (V, E)$, where node partitions correspond to DNA sequences. Each partition comprises $N = N_0 - l + 1$ vertices associated with all possible windows of length $l$. Thus, the overall number of vertices is $Np$. For every pair of vertices $u$ and $v$ in different partitions, an edge $(u, v) \in E$ is defined. The weight $w_{uv}$ assigned to an edge $(u, v)$ is proportional to the number of different bases between two subsequences corresponding to vertices $u$ and $v$. Thus, the objective of the motif finding problem reduces to choosing a node from each partition to minimize the weight of the corresponding induced subgraph. Note that the problem is equivalent to the maximum-weighted clique problem over
a $p$-partite graph. Fig. 1 illustrates the graphical model of an example with length-2 motif over three DNA sequences {ATG, CAT, TAT}.

![Graphical model of the motif-finding problem](image)

**Fig. 1.** Graphical representation of the motif-finding problem.

Using the graphical model described above, the motif finding problem is reformulated in [1] as an integer linear programming (ILP). For the $p$-partite graph, a binary variable $x_u$ and a binary variable $\gamma_{uv}$ are introduced for each vertex $u$ and for each edge $(u, v)$, respectively. Then, the ILP problem is expressed as

$$
\max \sum_{(u,v) \in E} w_{uv} \gamma_{uv}
$$

subject to

Type I: \[ \sum_{u \in V_j} x_u = 1 \text{ for } 1 \leq j \leq N \]

Type II: \[ \sum_{u \in V_j} \gamma_{uv} = x_v \text{ for } 1 \leq j \leq N, \]

where $v \in V \setminus V_j$, $u \in V$, $(u, v) \in E$. The first and the second constraints in (1) are referred to as ‘Type I’ and ‘Type II’ constraints, respectively. The formulation (1) is represented graphically in Fig. 2. The ‘vertex’ and ‘edge’ variable nodes in Fig. 2 are associated with vertex variables $x_u$ and edge variables $\gamma_{uv}$, respectively. Furthermore, the constraint nodes in Fig. 2 impose ‘Type I’ and ‘Type II’ constraints of (1).

Straightforward application of the message-passing framework to the model in Fig. 2 leads to the algorithm presented in [2]. However, careful examination of the graphical model in Fig. 2 reveals that this simple algorithm fails to converge for large-scale problems. Namely, each vertex node has only one adjacent ‘Type I’ constraint node and $p - 1$ adjacent ‘Type II’ constraint nodes. For problems with large $N$ and $p$, by the law of large number, the sum of incoming messages from ‘Type II’ constraint nodes at each vertex node is similar to that of other vertex nodes in the same partition. Therefore, the outgoing message from the ‘Type I’ constraint node to each vertex node is of similar value and the convergence of the vertex node beliefs is very slow. To address these shortcomings, we develop an alternative graphical model and use it to derive algorithms with improved convergence properties.

### 3. DESCRIPTION OF THE ALGORITHMS

To simplify the graphical model, we merge each vertex node and its adjacent ‘Type II’ constraint node into a new constraint node; the new constraint nodes in each partition are grouped and treated as a ‘super’ node. A closer look into the constraints reveals that the ‘Type I’ constraints are now implicitly imposed and can be removed. The corresponding graphical model is shown in Fig. 3. In Fig. 3, oval objects containing new constraint nodes denote the ‘super’ nodes. Only one constraint node from each ‘super’ node can be chosen in the final solution, which essentially enforces the ‘Type I’ constraint. Note that the number of unknown variables is now reduced to the number of the edge variables.

![Graphical model for the simplified message-passing](image)

**Fig. 3.** Graphical model for the simplified message-passing.

Having introduced the new graphical model above, we need to define messages over the edges of the graph and specify the algorithm over the messages. Note that combinatorial inference problems may have multiple solutions. Membership of each edge to the maximum-weighted clique (i.e., to the solution to the motif finding problem) is expressed in probabilistic terms. In particular, we introduce the marginal distribution for the state of each edge by counting the number of solutions containing that edge within the set of possibly multiple solutions. We choose the messages between two constraint nodes $i$ and $j$ in different partitions to be the log-likelihood ratio (LLR) of the marginal distribution of the edge $(i, j)$ contained in the optimal solution (which are denoted by $r^i_{t \to j}$ and $r^j_{t \to i}$ for two directions, respectively), i.e. the loga-
rithm of the ratio of the probability of being contained to that of not being contained. Let $p_{j \rightarrow i}(1)$ denote the probability that the edge $(i, j)$ is contained in the optimal solution, and $p_{j \rightarrow i}(0)$ the probability that the edge $(i, j)$ is not contained in the optimal solution. For convenience, we define the log-likelihood of above probabilities, $m^i_{j \rightarrow i}(1) \equiv \log p^1_{j \rightarrow i}(1)$, $m^i_{j \rightarrow i}(0) \equiv \log p^0_{j \rightarrow i}(0)$, and $r^i_{j \rightarrow i} \equiv m^i_{j \rightarrow i}(1) - m^i_{j \rightarrow i}(0)$. Let $P_i$ and $P_j$ denote the partitions which contain the variables $i$ and $j$, respectively. The messages which converge to the above defined marginal probabilities can be computed as

$$p^i_{j \rightarrow i}(1) = \exp(w_{ij}) \prod_{i' \neq i \in P_i} \left( \sum_{i'' \neq i' \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right),$$

$$p^i_{j \rightarrow i}(0) = \left( \sum_{i' \neq i \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right) \times \left( \sum_{i' \neq i \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right).$$

Log-likelihoods of the messages in (2) can be expressed as

$$m^i_{j \rightarrow i}(1) = w_{ij} + \sum_{i' \neq i \in P_i} \log \left( \sum_{i'' \neq i' \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right),$$

$$m^i_{j \rightarrow i}(0) = \log \left( \sum_{i' \neq i \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right) + \sum_{i' \neq i \in P_i} \log \left( \sum_{i'' \neq i' \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right).$$

The above expressions define a new iterative algorithm (essentially, the sum-product algorithm), which can further be simplified by using the LLR representation and related approximation techniques. The LLR of the messages in (3) is

$$r^i_{j \rightarrow i} = m^i_{j \rightarrow i}(1) - m^i_{j \rightarrow i}(0) = w_{ij} + \sum_{i' \neq i \in P_i} \log \left( \sum_{i'' \neq i' \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right) - \log \left( \sum_{i' \neq i \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right).$$

We approximate the logarithm of the sum of exponential functions by the max function (this is a standard step when deriving a min-sum algorithm from a sum-product algorithm),

$$r^i_{j \rightarrow i} = w_{ij} + \sum_{i' \neq i \in P_i} \log \left( \sum_{i'' \neq i' \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right) - \log \left( \sum_{i' \neq i \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right).$$

Thus, the final update rule is given by the simple form in (5).

We will refer to this algorithm as ‘Algorithm A.’ Note that the update rule is of the form similar to that of the message-passing algorithm for multiple matching problem [3] where there are $p$ partitions of nodes to be matched.

On the other hand, note that we are only interested in edges which are contained in the optimal solution. Therefore, we may ignore the log-likelihoods of edges not contained in the solution by setting them to zero, which leads to the following algorithm,

$$r^i_{j \rightarrow i} = m^i_{j \rightarrow i}(1) - m^i_{j \rightarrow i}(0) = w_{ij} + \sum_{i' \neq i \in P_i} \log \left( \sum_{i'' \neq i' \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right) - \log \left( \sum_{i' \neq i \in P_i} \prod_{t' \in P_t} p_t^{i' \rightarrow i}(1) \prod_{t'' \in P_t} p_t^{i'' \rightarrow i}(0) \right).$$

We will refer to this algorithm as ‘Algorithm B.’ The final update rule in (6) coincides with the min-sum algorithm for MAP assignment over a network with two node interactions [5, 6]. The expressions (5) and (6) imply close relationship between the motif finding problem and the aforementioned network assignment problem. This relationship will be analyzed in the extended version of the current paper. To further improve the convergence properties of the algorithms, we can apply the message-passing technique to the dual formulation of the original ILP (1), as proposed in [4]. This, too, will be elaborated in the extended version of the paper. We omit any further the details for brevity and formalize the updating equations in the algorithm below:

1. **Initialization**: At times $t = 0, 1, \ldots$, each vertex sends real-valued messages to each of its neighbors. The message from $i$ to $j$ at time $t$ is denoted by $m_{i \rightarrow j}(t)$. Messages are initialized by $m_{i \rightarrow j}(0) = w_{ij}$ for all $(i, j) \in E$. 

1. **Initialization**
2. **Message updates**: Messages in the $t^{th}$ iteration, $t \geq 1$, are obtained recursively from the messages in the $(t - 1)^{th}$ iteration using one of the following two rules,

\[
\begin{align*}
    r_{j \rightarrow i}^t &= w_{ij} - \max_{i' \neq i, i' \in P_t} r_{i' \rightarrow j}^{t-1}, \\
    r_{j \rightarrow i}^t &= w_{ij} + \max_{P_h \neq \{P_t, P_t\}} \sum_{P_h \neq \{P_t, P_t\}} \max_{l \in P_h} r_{l \rightarrow i}^{t-1}.
\end{align*}
\]

3. **Motif estimation**: Calculate $b_i = \sum_{P_t \neq P_t, \max_{j \in P_t} r_{j \rightarrow i}^{(t)}}$, the belief for each variable $i \in P_t$ of each partition $P_t$. Estimate the motif by choosing $p$ variables $M^{(t)} = \{i_1, i_2, \ldots, i_p\}$, one from each partition.

4. **Iteration**: Set $t = t + 1$. Repeat 2-3 until $M^{(t)}$ converges.

4. **EXAMPLES AND CONCLUSION**

We test the performance of the proposed algorithm on data sets containing the \textit{S. cerevisiae} transcription factor RCS1. We search over the data sets for a single motif of length $l = 7$ in $p$ four-alphabet sequences of length $N_0$ ($8 \leq p \leq 20$, $20 \leq N_0 \leq 100$). We compare the two new algorithms with the algorithm in [2]. For moderate $N_0$, the algorithm in [2] can determine identical patterns in 70% of the simulation runs, and fails to obtain identical patterns in the rest. Among the simulation runs where the solution was found, in 90% of the cases the algorithm converges within 5 iterations; in the remaining 10% of the cases, it converges within 10 iterations. For large $N_0$, the algorithm in [2] determines the motif in only half of the simulation runs. On the other hand, Algorithm A proposed in this paper determines identical patterns for all values of $N_0$ within 3-4 iterations. However, Algorithm A sometimes oscillates between the correct pattern and a less meaningful pattern. In spite of this, for large $N_0$ Algorithm A converges more often than the algorithm in [2], and the converged solution is always correct. Finally, among the three algorithms, Algorithm B has the best performance and convergence properties. It determines the identical pattern in all simulation runs, converging within 1-2 iterations.

The total number of the nodes in the graph in Fig. 3 is $Np$. Since the number of operations at each node for both Algorithm A and Algorithm B is $N^2p$, their complexity is $O(N^2p^2)$. Note that the complexity of the algorithm in [2] is $O(N^4p^2)$. Figures 4 and 5 show the average running time of the algorithms as a function of $N$ and $p$, respectively. As a comparison, we should point out that the average complexity of the ILP algorithm [1] is exponential, $O(2^{N^2p^2})$.

In summary, starting from an existing graphical representation of the ILP formulation of the motif-finding problem, we introduced an equivalent and simpler graphical model. We used this model to derive two message-passing algorithms with improved performance and convergence properties compared to the existing solutions. Simulation results demonstrate that the new algorithms indeed outperform the previous solutions. Future work includes analytical study of the convergence properties of the proposed algorithms.

5. **REFERENCES**


