Combinatorial Optimization Approaches to Motif Finding

E. Zaslavsky
M. Singh*

Department of Computer Science & Lewis-Sigler Institute for Integrative Genomics,
Princeton University, Princeton, NJ 08544.

Abstract

Motivation: Discovering approximately repeated patterns, or motifs, in genomic sequences is an important problem in computational molecular biology. Most frequently, motif finding applications arise when identifying shared regulatory signals within DNA sequences or shared functional and structural elements within protein sequences. Due to the diversity of contexts in which motif finding is applied, several variations of the problem are commonly studied.
Results: We introduce a versatile combinatorial optimization framework for the motif finding problem, which couples graph pruning techniques with a novel integer linear programming formulation. Our method is flexible and robust enough to accommodate several variants of the motif finding problem, and we extend it to discover multiple motifs, as well as motifs found in evolutionarily related sequences of varying phylogenetic distance. In contrast to commonly-used stochastic search methods for the problem, our combinatorial approach yields optimal solutions. We apply our method to numerous DNA and protein datasets, as well as to synthetic data, and in all cases it performs very well, identifying either known motifs or motifs of high conservation.
Contact: msingh@princeton.edu

Introduction

Motif discovery, or the problem of finding approximately repeated patterns in unaligned sequence data, is an important and long-studied problem in computational molecular biology. It has applications in uncovering transcriptional networks, as short common subsequences in the data may

[*] To whom correspondence should be addressed.

correspond to a regulatory protein's binding sites, and in protein function identification, where short blocks of conserved protein sequences code for important structural or functional elements.

Numerous approaches to motif finding have been suggested (e.g., [Lawrence and Reilly 1990, Lawrence *et al.* 1993, Bailey and Elkan 1995, Brazma *et al.* 1998, Rigoutsos and Floratos 1998, Hertz and Stormo 1999, Tompa 1999, Hughes *et al.* 2000, Marsan and Sagot 2000, van Helden *et al.* 2000, Workman and Stormo 2000, Pevzner and Sze 2000, Liu *et al.* 2001, Eskin and Pevzner 2002, Buhler and Tompa 2002, Sinha and Tompa 2003, Pavesi *et al.* 2004, Frith *et al.* 2004]). The biological problems addressed by motif finding are complex and varied, and no single currently existing method can solve them completely (e.g., see [Tompa *et al.* 2005]). For DNA sequences, motif finding algorithms have typically been applied to sets of sequences from a single genome that have been identified as possessing a common motif, either through DNA microarray studies [Tavazoie *et al.* 1999], ChIP-chip experiments [Lee *et al.* 2002] or protein binding microarrays [Mukherjee *et al.* 2004]. An orthogonal approach, which attempts to identify regulatory sites among a set of orthologous genes across genomes of varying phylogenetic distance, is adopted by [McGuire *et al.* 2000, McCue *et al.* 2001, Blanchette and Tompa 2002, Kellis *et al.* 2003, Cliften *et al.* 2003]. Yet another formulation of motif finding for DNA sequences, that of 'subtle' motifs, was introduced by [Pevzner and Sze 2000]. For protein sequences, motif finding can reveal structural and functional constraints, and especially in the case of divergent sequence motifs, incorporating amino acid substitution matrices [Dayhoff *et al.* 1978, Henikoff and Henikoff 1992] is particularly useful.

Here, we consider a combinatorial optimization framework for motif finding that is flexible enough to model several variants of the problem. Underlying our approach, we consider motif finding as the problem of finding the best gapless local multiple sequence alignment using the sum-of-pairs (SP) scoring scheme, and then give extensions to account for multiple motifs and phylogenetic distance. The SP-score is one of many reasonable schemes for assessing motif conservation [Osada *et al.* 2004, Schuler *et al.* 1991, Carillo and Lipman 1988]. We then formulate the motif finding problem as an instance of integer linear programming (ILP), which is NP-hard to solve in general, and consider its linear programming (LP) relaxation. Typically, the linear programs are very large, numbering in the millions of variables, and prove too difficult even for highly optimized commercial solvers. To reduce the complexity of the linear programs, we employ a number of pruning techniques, building upon the ideas of [Gusfield 1993, Vingron and Pevzner 1995, Pevzner and Sze 2000, Lukashin and Rosa 1999]. These fall into the broad category of dead-end elimination (DEE) algorithms (e.g. [Desmet *et al.* 1992]), where sequence positions that are incompatible with the optimal solution are discarded. The resulting linear programs are often small, and are easily solved by the solver [CPLEX 7.1]. Interestingly, the vast majority of solutions are integral, guaranteeing optimality for the original integer linear program and motif finding problem, and obviating the need to employ ILP solvers. Thus, our approach runs in polynomial time for many practical instances of the problem. Since it is known that finding optimal global or local multiple sequence alignments under the SP measure is NP-hard [Akutsu *et al.* 2000, Wang and Jiang 1994], integral solutions to the linear programs and consequently polynomial-time convergence of the algorithm cannot be guaranteed. In the cases where fractional solutions are found for the linear programs, integer linear programming solvers are applied to find optimal solutions. In practice, the ability of our method to find optimal solutions to large problems attests to its overall effectiveness.

We test our coupled mathematical programming and pruning approach in various settings. First, we consider the problem of finding shared sequence motifs in protein sequences. Unlike the commonly-used stochastic search methods for motif finding (e.g., [Lawrence *et al.* 1993, Bailey and Elkan 1995], our combinatorial formulation naturally incorporates amino acid substitution matrices, and also guarantees optimal solutions for the objective function being maximized. Second, we consider sets of genes known to be regulated by the same *E. coli* transcription factor, and apply our approach to find the corresponding binding sites. Third, we consider the phylogenetic footprinting problem [Blanchette and Tompa 2002], and find shared motifs upstream of orthologous genes. The difficulty of this problem lies in that the sequences may not have had enough evolutionary time to diverge and may share sequence level similarity beyond the functionally important site; incorporation of additional information, in the form of the weightings obtained from a phylogenetic tree relating the species, proves useful in this context. Finally, we consider the subtle motifs formulation of [Pevzner and Sze 2000], where a fixed pattern is inserted into the input sequences with some number of perturbations. In particular, we show that our formulation can be used to find many optimal solutions, thereby retrieving the correct implanted one, along with others that may occur by chance. In all scenarios, we show that our method works well in practice, either recovering the known motifs or other motifs of high conservation.

Methods

Broad Problem Formulation

The motif finding problem is modeled here as that of finding the ungapped local multiple sequence alignment (MSA) with best sum-of-pairs (SP) score. Informally, given N sequences $\{S_1, \ldots, S_N\}$ and a block length parameter l, the goal is to find an l-long subsequence from each input sequence so that the total similarity among selected blocks is maximized. More formally let s_i^k refer to the l–long block (l-mer) in sequence S_i beginning in position k and let $sim(x, y)$ denote a similarity score between the l-long subsequences x, y. The objective is then to find the set of positions $\{k_1, \ldots, k_N\}$ in each sequence, such that the sum-of-pairs score $\sum_{i<j} sim(s_i^{k_i}, s_j^{k_j})$ is maximized.

It is convenient to consider a graph-theoretic formulation of this problem [Reinert *et al.* 1997]. Let G be an undirected N-partite graph with node set $V_1 \cup \ldots \cup V_N$, where V_i includes a node u for each l-long subsequence s_i^k in the i-th sequence. Note that the subsequences corresponding to two consecutive vertices overlap in $l-1$ positions, and that the V_i's may have varying sizes. Each pair of nodes $u \in V_i$ and $v \in V_j$ $(i \neq j)$, corresponding to subsequences s_i^k and $s_j^{k'}$ in S_i and S_j respectively, is joined by an edge with weight of $w_{uv} = sim(s_i^k, s_j^{k'})$. By this construction G is a complete N-partite graph. The MSA is achieved by picking the highest weight N-partite clique (denoted N-clique) in graph G.

The rest of this section describes the combinatorial optimization framework for the MSA problem. We first explain our approach to the basic formulation of the problem, and then consider extensions. Our method consists of two components: the mathematical programming formulation of the maximum-weight clique problem in G, and graph pruning techniques. Though the problem can theoretically be solved using mathematical programming tools directly, biologically relevant instances are typically very large, numbering in the millions of variables, and would take a prohibitively long time to solve. To reduce the running-times we employ a number of pruning techniques, generally referred to as dead-end elimination (DEE) in the protein design community, which discard vertices and/or edges that cannot possibly be part of the optimal solution[1].

[1] Our framework can be recast as a minimization problem as well. In that case the graph edge weights are derived from a distance measure between l-long subsequences. The MSA is then achieved by picking the lowest-weight clique in the graph, and the described DEE techniques can be easily adjusted.

Basic Motif Finding Framework

Similarity scores

To fully specify the graph as above, we need to define its edge weights. In the simplest case of finding DNA motifs we use a 1/0 similarity score for match/mismatch between pairs of bases, and sum the scores for l–long blocks in pairs of sequences to derive the weights. We also apply the same basic problem formulation to the protein motif finding problem. In this case we compute the weights based on amino acid substitution matrices, which assign higher scores to more favorable substitutions and better reflect biochemical properties of such pairings. We experiment with both PAM [Dayhoff *et al.* 1978] and BLOSUM [Henikoff and Henikoff 1992] matrix families. To calculate the edge weights, we sum the matrix entries for amino acid pairs in each position of the l-long block.

Integer Linear Programming Formulation

For graph $G = (V, E)$, where $V = V_1 \cup \ldots \cup V_N$ and $E = \{(u, v) : u \in V_i, v \in V_j, i \neq j\}$, we introduce a binary decision variable x_u for every vertex u, and a binary decision variable y_{uv} for every edge (u, v). Setting x_u to 1 corresponds to selecting vertex u for the N-clique and thus choosing the sequence position corresponding to u in the alignment. Setting variable y_{uv} to 1 corresponds to choosing both the vertices u and v for the N-clique.

The following integer linear program solves the motif finding problem formulated above:

$$\begin{array}{llll}
\text{Maximize} & \sum_{(u,v)\in E} w_{uv} \cdot y_{uv} & & \\
 & & & \\
\text{subject to} & \sum_{u\in V_j} x_u = 1 & \text{for } 1 \leq j \leq N & (\textit{node} \text{ constraints}) \\
 & \sum_{u\in V_j} y_{uv} = x_v & \text{for } 1 \leq j \leq N, v \in V \setminus V_j & (\textit{edge} \text{ constraints}) \\
 & x_u, y_{uv} \in \{0, 1\} & \text{for } u \in V, (u, v) \in E &
\end{array}$$

The first set of constraints ensures that exactly one vertex is picked from every graph part, corresponding to one position being chosen from every input sequence. The second set of constraints relates vertex variables to edge variables, allowing the objective function to be expressed in terms of finding a maximum edge-weight clique. An edge is chosen only if it connects two chosen vertices. This formulation is similar to that used by [Kingsford *et al.* 2005] for fixed-backbone protein design and homology modeling.

ILP itself is NP-hard, but replacing the integrality constraints on the x and y variables with $0 \leq x_u, y_{uv} \leq 1$ gives a polynomial-time heuristic for the problem. It is important to note that should a linear programming solution happen to be integral, it is guaranteed to be optimal for the original ILP and motif finding problem. Non-integral solutions, on the other hand, are not feasible for the ILP and do not translate to a selection of positions for the MSA problem. Those instances need to be solved by other means, such as using an ILP solver. Interestingly, we find integral solutions in an overwhelming majority of instances (especially after applying our pruning techniques).

Graph pruning techniques

Basic clique-bounds DEE. The idea of our first pruning technique is as follows. Suppose there exists a clique of weight C^* in G. Then a vertex u, whose participation in any possible clique in G reduces the weight of that clique below C^*, is incompatible with the optimal alignment and can be safely eliminated (similar to [Lukashin and Rosa 1999]).

For $u \in V_i$ define $star(u)$ to be a selection of vertices from every graph part other than V_i. Let F_u be the value induced by the edge weights for a $star(u)$ that form best pairwise alignments with u:

$$F_u = \sum_{j \neq i} \max_{v \in V_j} w_{uv} \tag{1}$$

If u were to participate in any clique in G, it cannot possibly contribute more than F_u to the weight of the clique. Similarly, let F_i^* be the value of the best possible $star(u)$ among all $u \in V_i$:

$$F_i^* = \max_{u \in V_i} F_u \tag{2}$$

F_i^* is an upper bound on what any vertex in V_i can contribute to any alignment.

Now, if F_z, the most a vertex $z \in V_k$ can contribute to a clique, assuming the best possible contributions from all other graph parts, is insufficient compared to the value C^* of an existing clique, i.e. if

$$F_z < 2 \times C^* - \sum_{i \neq k} F_i^*, \tag{3}$$

z can be discarded. The clique value C^* is used with a factor of 2 since two edges are accounted for between every pair of graph parts in the above inequality.

In fact, the values of F_i^* are further constrained by requiring a connection to z when z is under consideration. That is, when considering a node $z \in V_k$ to eliminate, and calculating F_i^* according to Equation 2 among all possible $u \in V_i$, the F_u of Equation 1 is instead computed as:

$$F_u = w_{zu} + \sum_{j \neq i,k} \max_{v \in V_j} w_{uv} \tag{4}$$

The value of C^* can be computed from any "good" alignment. We use the weight of the clique imposed by the best overall *star*.

Tighter constraints for clique-bounds DEE. For a vertex $u \in V_i$ and every other V_j, an edge has to connect u to some $v \in V_j$ in any alignment. When calculating F_u, we can constrain its value by considering three-way alignments and requiring that the vertices in the best $star(u)$ chosen as neighbors of u in graph parts other than V_j are also good matches to v. Performing this computation for every pair of u, V_j and considering every edge incident on u would be too costly. Therefore, we only consider such three-way alignments for every vertex u in the graph where $u \in V_i$ and the next part V_{i+1} of the graph (with the last and first parts paired). Essentially, this procedure shifts the emphasis onto edges, allowing better alignments and bounds, and yet eliminates vertices by considering the best edge incident on it. We define F_{uv} for edge (u, v) with endpoints $u \in V_i$ and $v \in V_{i+1}$ as

$$F_{uv} = w_{uv} + \frac{1}{2} \sum_{\substack{j \neq i \\ j \neq i+1}} \max_{x \in V_j} (w_{ux} + w_{vx}). \tag{5}$$

F_{uv} can be viewed as summing over two *stars*, one centered at u and the other at v. Defining F_u for $u \in V_i$ as

$$F_u = \max_{v \in V_{i+1}} F_{uv}.$$

and leaving the other definitions as above, Equation 3 can be used to eliminate vertices. The scaling above is necessary as every pair of graph parts i, j would have otherwise been accounted for four times[2] in Equation 3.

Graph Decomposition.DEE techniques work well for simpler instances of the motif finding problem (see Results), but tend to be inadequate, leaving a large final graph, for more complex cases. To overcome this difficulty we propose a divide-and-conquer graph decomposition approach. For every graph part i and vertex $u \in V_i$ we consider induced subgraphs $G^u = (V^u, E^u)$ in turn, where $V^u = u \ \cup \ V \setminus V_i$. Application of the *clique-bounds* DEE technique to graphs G^u is very effective

[2] Pairs of adjacent parts $i, i+1$ would have been counted once directly as the first term of Equation 5 and twice when considering pairs $i-1, i$ and $i+1, i+2$. Pairs of non-adjacent parts i, j would have been counted once each when considering adjacent parts $i-1$ and i, i and $i+1$, $j-1$ and j, and j and $j+1$.

since one of the graph parts, G_i^u contains only one vertex, u, and all the F and F^* values that need to be recomputed for the new graph G^u are greatly constrained. The process of updating the F and F^* values is efficient as the changes are localized to one part in the graph. Importantly, the C^* remains intact, since the clique of that larger value exists in the original graph and can be used for the decomposed one, helping to eliminate vertices. For some of the vertices u, iterative application of the DEE criterion and re-computation of the F and F^* values causes G^u to become disconnected, implying that vertex u cannot be part of the optimal alignment. Such a vertex u is marked for deletion, and that information is propagated to all subsequently considered induced subgraphs, further constraining the corresponding F and F^* values and helping to eliminate other vertices in turn.

Algorithm Description

We combine the various elements described above and apply them in the order of increasing complexity. At every juncture a decision is made whether to send the problem in its current state to the LP solver, and this decision hinges on the size of the graph. If the graph is "small" enough for some suitable definition of small (currently set at 600 vertices), we submit the appropriate linear program to the LP solver and, in rare instances, to the ILP solver. To reduce the graph to that necessary small size, we apply the DEE variants, terminating the process when the specified size has been reached. First, we attempt to prune the graph using *basic clique-bounds* DEE alone, and then consider *graph decomposition* in conjunction with *basic clique-bounds* DEE and tighter bound computations.

Subtle Motifs Framework

Pevzner and Sze [Pevzner and Sze 2000] introduced the 'subtle' motifs version of the motif finding problem. They formulate it as a signal finding problem in which an unknown pattern of a given length is inserted with modifications into each of the input sequences. The positions of insertion are unknown, as are the modifications in the instances of the pattern. Pevzner and Sze focus on what they call the (l, d)-signal version, in which the pattern is a string of length l and each pattern instance differs from the pattern in exactly d positions. The mutations are allowed to occur anywhere in the pattern, and thus any two instances of the pattern may differ from each other in as many as $2d$ positions. On the other hand, if two subsequences differ in more than $2d$ positions, they cannot possibly be instances of the implanted pattern.

We can solve the subtle motifs problem just as above by formulating it as a multiple sequence alignment with the sum-of-pairs score. The graph version of the problem remains largely the same except that it is no longer a complete N-partite graph. By definition, vertices that correspond to subsequences whose Hamming distance is greater than $2d$ should not be connected by an edge, as such an edge cannot possibly be part of the optimal clique. The weights on the edges, as suggested in [Pevzner and Sze 2000], are computed by considering the number of matches and mismatches.

It is straightforward to adjust our linear program to reflect the fact that graph G is no longer a complete graph by removing variables corresponding to non-existent edges. The main difference is in the *edge* constraints, in that the summation is made over the existing edges only.

Graph pruning. We begin by noting that these graphs may be pruned using any of the methods introduced by previous authors (e.g., in [Pevzner and Sze 2000, Sze *et al.* 2004]), and our LP/ILP can be applied whenever the graph size has decreased sufficiently. Here, we experiment with only a couple of DEE procedures.

The first DEE technique we apply is that of *graph connectivity*, suggested in [Vingron and Pevzner 1995, Pevzner and Sze 2000]. Since graph $G = (V, E)$ is no longer a complete N-partite graph, we can use connectivity properties of G to prune "dead-end" vertices and edges. Following the notation of [Pevzner and Sze 2000], let vertex $u \in V_i$ be a *neighbor* of vertex $v \in V_j$ if (u, v) is an edge in the graph, and vertex x be a *neighbor* of an edge (u, v) if $\{u, v, x\}$ is a connected triangle in the graph. The strategy we call *vertex removal* is to delete any vertex u that does not have at least one neighbor in every part of G other than V_i. *Edge removal* eliminates any edge (u, v) that does not have at least one neighbor in every part of G excluding V_i and V_j. These two strategies are applied iteratively until no further vertices or edges are detected for removal.

We also introduce a second *graph decomposition* technique for solving more difficult instances of the subtle motifs problem. Though any DEE routine can be employed in conjunction with graph decomposition, we concentrate on the iterative application of *vertex removal* and *edge removal* procedures. The idea of decomposition becomes more effective if modified slightly from its version above. Here we choose an arbitrary part V_i of V, and consider all the vertices $u \in V_i$ in turn. The intuition is that some vertex u in V_i has to be in the optimal alignment, and considering each one exhausts the possibilities. As before, we consider induced subgraphs $G^u = (V^u, E^u)$ and process them with the connectivity DEE routines. For some u, their corresponding graphs G^u become disconnected, and those vertices can be discarded; for others, typically, the graph remaining after this processing is small. As a final step we solve a number of these small linear programs, and select the optimal solution to the original problem among them. Note that for some difficult and dense instances of the subtle motifs problem, we can extend this *graph decomposition* technique to multiple graph parts, considering pairs or even triplets of vertices for elimination.

Other Motif Finding Frameworks

Phylogenetic Footprinting

As mentioned earlier, one way of finding regulatory sites is to look for them among a set of homologous genes across species. In this case additional data, in the form of the phylogenetic tree relating the species, is available and should be exploited. It is especially important when closely related species are part of the input, and, unweighted, they contribute duplicate information and skew the alignment. We use a phylogenetic tree and branch lengths when calculating the edge weights in the graph, with highly diverged sequence pairs getting larger weights. The precise weighting scheme follows the ideas of weighted progressive alignment [Feng and Doolittle 1987], in which weights α_i are computed for every sequence i. The calculation sums branch lengths along the path from the tree root to the sequence at the leaf, splitting shared branches among the descendant leaves, and thereby reducing the weight for related sequences. In essence, we solve a multiple sequence alignment problem with weighted SP-score using match/mismatch, where the computed weight for a pair of positions in sequences i and j is multiplied by $\alpha_i \times \alpha_j$. The rest of the algorithm operates as in the basic motif finding case above, employing the same linear programming formulation and DEE techniques.

Multiple Motifs

Here we give several extensions to address the issue of multiple motifs existing in a set of sequences. First, distinct multiple motifs, such as sets of binding sites for two different transcription factors, can be found iteratively by first locating a single optimal motif, masking it out from the problem instance, and then looking for the next one. We mask the previous motif by deleting its solution vertices from the original graph, and then reapplying the DEE/LP techniques to locate the next optimal motif.

Second, it is possible to solve iteratively several ILPs in order to find multiple near-optimal solutions, corresponding to the best cliques of successively decreasing total weights. At iteration t, we add a constraint to the integer linear programming formulation so as to exclude all previously discovered solutions:

$$\sum_{u \in S_k} x_u \leq N - 1 \quad \text{for } k = 1, \ldots, t-1, \tag{6}$$

where S_k contains the optimal set of vertices found in iteration k. This requires that the new solution differs from all previous ones in at least one graph part. We note that to use this constraint for the basic formulation of the motif finding problem, the DEE methods given above have to be modified so as not to eliminate nodes taking part in near-optimal but not necessarily optimal solutions. For the subtle motifs problem, the DEE methods only eliminate nodes and edges based on whether they can take part in any clique in the graph, and thus constraint 6 can be immediately applied to iteratively find all possible cliques.

Finally, finding *repeated* motifs where a single pattern occurs in m distinct positions in each sequence, requires a slight modification to the construction of the graph $G = (V, E)$. The set of vertices remains the same, but the edge set is modified in two ways. First, we introduce edges between vertices corresponding to positions in the same sequence, since the instances of the motif are all mutually similar. Secondly, to address the issue of low complexity regions, such as poly-A repeats, being selected as a solution, we would like the motif instances to be non-overlapping. Thus, vertices corresponding to overlapping l-mers in each sequence are not connected by an edge. The ILP formulation to address this problem is similar to the case of single motif; additional constraints are needed, though, to ensure a proper vertex selection since the graph now has edges within each part.

$$\begin{array}{llll}
\text{Maximize} & \sum_{(u,v)\in E} w_{uv} \cdot x_{uv} & & \\
 & & & \\
\text{subject to} & \sum_{u\in V_j} x_u = m & \text{for } 1 \leq j \leq N & (\textit{node} \text{ constraints}) \\
 & \sum_{u\in V_j} y_{uv} = m \times x_v & \text{for } 1 \leq j \leq N, v \in V \setminus V_j & (\textit{inter-part edge} \text{ constraints}) \\
 & \sum_{u\in V_j} y_{uv} = (m-1) \times x_v & \text{for } 1 \leq j \leq N, v \in V_j & (\textit{intra-part edge} \text{ constraints}) \\
 & x_u, y_{uv} \in \{0,1\} & \text{for } u \in V, (u,v) \in E &
\end{array}$$

The difference with the single motif ILP in the *node* constraints is in that m of the vertices are chosen in each graph part, corresponding to m positions in each input sequence. The next two sets of constraints ensure a proper edge selection between chosen vertices both inside and between partitions.

Experimental Analysis

We apply the basic combinatorial optimization framework to several motif finding problems. We attempt to discover motifs in instances arising from both DNA and protein sequence data. We then discuss subtle motif finding in simulated data.

Protein motifs

We study the performance of our algorithm on a number of protein datasets with different characteristics. The datasets, summarized in Table 1, were constructed using the SwissProt [Boeckmann *et al.* 2003] database from the descriptions of [Lawrence *et al.* 1993,Lukashin and Rosa 1999,Neuwald *et al.* 1995]. These datasets are highly variable in the number and length of their input sequences as well as the degree of motif conservation. The motif length parameters are set based on the lengths described by the above authors. The default amino acid substitution matrix we use for all the datasets is BLOSUM62. For more closely related proteins, like the human tumor necrosis factor (TNF) proteins, we also experiment with the BLOSUM80 and PAM100 matrices. However, choice of substitution matrix and slight variations in sought motif length do not substantially affect the results as similar motifs are found. The five datasets are of varying difficulty to solve, with some employing the basic *clique-bounds* DEE technique to prune the graphs, while others require more elaborate pruning that is constrained by three-way alignments (see Table 1).

In all the test datasets our algorithm recovers the motifs found by [Lawrence *et al.* 1993,Lukashin and Rosa 1999, Neuwald *et al.* 1995] and reported in the literature. As described by [Lawrence *et al.* 1993], the HTH dataset is very diverse, and the detection of the motif is a difficult task. Nonetheless, our HTH motif is identical to that of [Lawrence *et al.* 1993], and agrees with the known annotations in every sequence. We likewise find the lipocalin motif; it is a weak motif with few generally conserved residues that is in perfect correspondence with the known lipocalin signature. We also precisely recover the immunoglobulin fold, TNF and zinc metallopeptidase motifs. In contrast to [Lukashin and Rosa 1999], who limit sequence lengths to 500, we retain the original protein sequences, making the problem more difficult computationally as the average sequence length in the zinc metallopeptidase dataset is approximately 800, and some sequences are as long as 1300 residues[3]. Nonetheless, we find the motif using our method, and some of the motif

[3] Our implementation of the method of [Lukashin and Rosa 1999] applied to the zinc metallopeptidase dataset with full length sequences failed to converge.

instances we recover are superior to those of [Lukashin and Rosa 1999].

DNA motifs

We analyze the performance of our method on a set of sequences consisting of DNA binding sites embedded in their respective upstream regions (up to 600 bp) for a number of regulatory proteins. Our dataset is constructed from the data of [Robison *et al.* 1998, McGuire *et al.* 2000] as described in [Osada *et al.* 2004]. In short, we remove duplicate sites, sigma factors and transcription factors with fewer than three known binding sites. Additionally, we include only one copy of a sequence corresponding to multiple known binding sites. Of the 35 transcription factor families we considered, 23 were solved in seconds with the application of *basic clique-bounds*, 10 required application of *graph decomposition* with *clique-bounds* DEE, constrained by three-way alignments and took a few minutes to three hours to solve, and the two largest datasets, CRP and IHF, were not reduced in size by our DEE methods and their linear programs proved infeasible to solve. Of the 33 solved problems all but two resulted in integral solutions to their linear programs that immediately translated to a motif selection. The two instances with fractional solutions were easily solved by the ILP solver.

Evaluating performance of any motif finding algorithm is not a straightforward task, as other, better-conserved, biologically-relevant motifs may exist in the data. To that end, we compute the degree of agreement between the motifs discovered by our method, and the known binding sites, and also compare them with motifs found by a widely used stochastic-search motif finder, Gibbs Motif Sampler [Thompson *et al.* 2003]. First we compare the quality of the discovered motifs by their average information content (IC), a measure related to one maximized by the Gibbs algorithm. Even though the measure we maximize is not directly related to IC, our approach finds motifs of slightly higher level of conservation as measured by average IC than the Gibbs Motif Sampler, identifying such better conserved motifs in 10 instances (vs. 7 better motifs by Gibbs Motif Sampler), and discovering an identical motif for 16 other instances. All motifs found by both methods exhibit equal or higher average IC than that of the sets of known transcription factor binding sites. Whereas our method fails to identify a motif in two datasets, the Gibbs Motif Sampler fails for five transcription factors (one dataset resulting in failure is shared), reporting no significant motif after 20 random restarts in each case. Surprisingly, of these five families two possess a motif of very high IC.

To determine the extent of agreement between the motif predictions versus known motifs, we compute the degree of overlap by a commonly used comparison statistic, proposed by [Pevzner and

Sze 2000], the *performance coefficient*, defined as follows. In a dataset of t sequences and motif length l, let P be the set of $t \times l$ sequence positions occupied by one (known) motif, and K be the set of $t \times l$ sequence positions occupied by the predicted motif. Then the performance coefficient ($p.c.$) is $|K \cap P|/|K \cup P|$. We give a histogram of the performance coefficients for our algorithm vs. Gibbs Motif Sampler in Figure 1. Though the methods perform similarly for the cases of extensive overlap where the biological motif has essentially been completely identified, overall our method discovers motifs that exhibit a slightly higher overlap with the known transcription factor regulatory sites, such that the average $p.c.$ for the entire dataset is 0.422 vs. 0.397 for the Gibbs Motif Sampler.

Phylogenetic Footprinting

We experiment with motif discovery among sets of upstream regions of orthologous genes in a number of genomes, having incorporated phylogenetic information in constructing our graphs. Here we use data sets, varied in size and genome selection, from [Blanchette and Tompa 2002]. We identify well-conserved interesting motifs in every dataset[4]. The consensus sequences for the discovered motifs are listed in Table 2 along with the description of their DNA regions and source species. All the motifs we find have been documented in the TRANSFAC database [Wingender *et al.* 1996], and the majority of them correspond to those that have been reported by [Blanchette and Tompa 2002].

This dataset is also an excellent testing ground for finding distinct multiple motifs using our method, as such motifs exist and have been reported in previous studies. We iteratively identify motifs and remove their corresponding vertices from the constructed graphs. As proof of principle, we find multiple motifs for the insulin dataset. In this case, we successfully identify all four motifs reported by [Blanchette and Tompa 2002]. Since our objective function differs from that of [Blanchette and Tompa 2002] and we require motif occurrences in every input sequence, we recover the motifs in a different order. Of course, we identify numerous shifts of motifs found in previous iterations before arriving at the next distinct motif. In practice, therefore, it may be more beneficial to remove all vertices corresponding to subsequences overlapping the optimal solution when modifying the problem for successive iterations.

[4] Motif reported for the C-fos promoter dataset was discovered second, after having discarded the poly-A repeat region.

Subtle motifs

We test our algorithm's performance in finding subtle motifs on synthetically generated data. Following the terminology of [Pevzner and Sze 2000], we produce the problem instances according to the *FM* (fixed mutation) model. For the (l, d)-signal finding problem we first randomly select a pattern M of length l. For each of t background sequences ($t = 20$ for all test cases) exactly d random positions are chosen in M, and the pattern is then implanted with each of these d positions mutated to a different, randomly chosen base. Both the background sequence and the position of the implanted pattern instance are selected randomly. All random choices above are independent and drawn uniformly.

The original challenge problem proposed by [Pevzner and Sze 2000] is the $(15, 4)$ motif finding problem with background sequence length 600. Various algorithms [Buhler and Tompa 2002,Keich and Pevzner 2002,Eskin and Pevzner 2002,Price *et al.* 2003] have extended the length to 1500 and beyond and considered other combinations of the l and d parameters. Previous approaches have reported the performance coefficient *p.c.* defined earlier, averaged over several generated problem instances (denoted *a.p.c.*) for every set of parameters considered.

Here, we implant $(15, 4)$, $(14, 4)$, and $(12, 3)$ patterns into background sequences of length $N = 600$. These problems are of varying difficulty; for example, the *a.p.c* for the Projection method of [Buhler and Tompa 2002] is 0.93 for $(15, 4)$, 0.71 for $(14, 4)$, and 0.77 for $(12, 3)$. We consider 20 random instances of each, and use constraint in Equation 6 to find up to the first 1000 heaviest weight cliques (if that many exist) by enumerating near-optimal solutions. In each case, one of the cliques output corresponds to the actual implanted pattern. Since edges between vertices are weighted by the total number of matches between the corresponding subsequences, the implanted motif typically corresponds to one of the higher scoring ones. For the simpler case of the $(15, 4)$ motif, the implanted pattern is often found among the optimal-weight cliques. However, in some of the $(14, 4)$ and $(12, 3)$ cases, there appear to be many higher weight cliques occurring by chance, suggesting that these patterns are too hard to distinguish from background. Nevertheless, our approach of finding multiple, successive near-optimal solutions allows us to retrieve all valid possible implanted patterns.

Discussion

We have described a versatile mathematical programming framework for the motif finding problem. In order to solve the linear programs, numbering sometimes in the millions of variables, we employ graph decomposition and pruning techniques to identify vertices guaranteed to be excluded from the optimal solution. While these algorithms cause a tremendous reduction to the size of the problem, some datasets are more challenging and computationally expensive than others. Clearly, the difficulty is correlated with the total number of vertices in the graphs, and the average sequence length is the dominant factor (see Table 1). Interestingly, and often regardless of problem size, most resulting linear programs exhibit integral solutions. The reasons for this phenomenon are unclear, and constitute an exciting research question. It is also noteworthy that presence of a well conserved motif allows the pruning to be especially effective. For example, the human zinc metallopeptidase dataset contains a very highly conserved motif, a well-recognized zinc-binding region signature; its reduced graph size is 10 (albeit with application of involved decomposition and pruning techniques), a mere one vertex per graph part.

A major advantage of our algorithm over previous approaches for motif finding is that we are able to find optimal local alignments for many practical problems. In the future, we hope to extend the capabilities of our approach by incorporating features common to more widely-used motif finding algorithms. A basic improvement would be to assess statistical significance of the discovered motifs; this is independent of the mathematical programming framework, and we hope to use some of the same techniques outlined in previous research (e.g., [Sinha and Tompa 2003]). Additionally, we would like to allow zero occurrences of a motif in some of the input sequence; this may be possible in our framework by including an additive term in the objective function that creates a tradeoff between the weight of the induced clique-like structure that is being maximized and the potential motif absence. It is possible to include other types of useful constraints into our linear program. For example, we may want to look for two shorter motifs that are within some distance of each other; there are natural linear constraints that can enforce this.

In summary, the described optimization framework provides a flexible approach to tackle many important issues in motif finding. We have successfully applied it to a variety of problems, including DNA motifs, protein motifs, and subtle motifs, and have been able to incorporate phylogenetic information in the context of cross-species motif discovery, as well as to find multiple near-optimal solutions. We hope in the future to extend its capabilities to model more complex types of motif finding problems.

References

[Akutsu *et al.* 2000] Akutsu, T., Arimura, H., and Shimozono, S. On approximation algorithms for local multiple alignment. In Proceedings of the Fourth Annual International Conference on Research in Computational Molecular Biology, pages 1–7. ACM Press, 2000.

[Bailey and Elkan 1995] Bailey, T. and Elkan, C. 1995. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine Learning, 21: 51–80.

[Blanchette and Tompa 2002] Blanchette, M. and Tompa, M. 2002. Discovery of regulatory elements by a computational method for phylogenetic footprinting. Genome Res. 12: 739–748.

[Boeckmann *et al.* 2003] Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M. 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. 31: 365–370.

[Brazma *et al.* 1998] Brazma, A., Jonassen, I,, Eidhammer, I., and Gilbert, D. 1998. Approaches to the automatic discovery of patterns in biosequences. J. Comput Biol. 5(2): 279–305.

[Buhler and Tompa 2002] Buhler, J. and Tompa, M. 2002. Finding motifs using random projections. J. Comput Biol. 9(2): 225–242.

[Carillo and Lipman 1988] Carillo, H. and Lipman, D. 1988. The multiple sequence alignment problem in biology. SIAM Journal on Applied Math. 48: 1073–1082.

[Cliften *et al.* 2003] Cliften, P., Sundarsanam P., Desikan, A., Fulton, L., Fulton, B., Majors, J. *et al.*. Finding functional features in Saccharomyces genomes by phylogenetic footprinting. 301: 71–76.

[CPLEX 7.1] ILOG CPLEX 7.1 http://www.cplex.com.

[Dayhoff *et al.* 1978] Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C. 1978. A model of evolutionary change in proteins. In "Atlas of Protein Sequence and Structure" 5(3) M.O. Dayhoff (ed.), 345–352.

[Desmet *et al.* 1992] Desmet, J., De Maeyer, M., Hazes, B., Lasters, I. 1992. The dead-end elimination theorem and its use in protein side-chain positioning. Nature 356: 539–542.

[Eskin and Pevzner 2002] Eskin, E. and Pevzner, P. Finding composite regulatory patterns in DNA sequences. 2002. Bioinformatics (Supplement 1), 18: S354–S363.

[Feng and Doolittle 1987] Feng, D. and Doolittle, R. F. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol., 60: 351–360.

[Frith *et al.* 2004] Frith, M.C., Hansen, U., Spouge, J.L. and Weng Z. 2004. Finding functional sequence elements by multiple local alignment. Nucleic Acids Res. 32(1): 189–200.

[Gusfield 1993] Gusfield, D. 1993. Efficient methods for multiple sequence alignment with guaranteed error bounds. Bull. Math. Biol. 55(1): 141–154.

[Henikoff and Henikoff 1992] Henikoff, S. and Henikoff, J. 1992. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. USA. 89(biochemistry): 10915–10919.

[Hertz and Stormo 1999] Hertz, G. and Stormo, G. 1999. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. Bioinformatics 15: 563–577.

[Hughes *et al.* 2000] Hughes, J., Estep, P., Tavazoie, S. and Church, G. 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. J. Mol. Biol. 296: 1205–1214.

[Keich and Pevzner 2002] Keich, U. and Pevzner, P. 2002. Finding motifs in the twilight zone. Bioinformatics, 18: 1374–1381.

[Kellis *et al.* 2003] Kellis, M. Patterson, N., Endrizzi, M., Birren, B. and Lander E. Sequencing and comparison of yeast species to identify genes and regulatory elements. Nature, 423: 241–254.

[Kingsford *et al.* 2005] Kingsford, C.L., Chazelle, B. and Singh, M. 2005. Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics 21(7): 1028–1039.

[Lawrence *et al.* 1993] Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A., and Wootton, J. 1993. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science 262: 208–214.

[Lawrence and Reilly 1990] Lawrence, C. and Reilly, A. 1990. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. Proteins: Structure, Fuction, and Genetics, 7: 41–51.

[Lee *et al.* 2002] Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., Young, R.A. 2002. Science 298(5594): 799–804.

[Liu *et al.* 2001] Liu, X., Brutlag, D.L., Liu, J.S. 2001. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. Pacific Symposium on Biocomputing, pages 127–138.

[Lukashin and Rosa 1999] Lukashin, A. and Rosa, J. 1999. Local multiple sequence alignment using dead-end elimination. Bioinformatics 15: 947–953.

[Marsan and Sagot 2000] Marsan, L., and Sagot, M. F. 2000. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. J. Comput Biol. 7(3-4): 3450–62.

[McCue *et al.* 2001] McCue, L., Thompson, W., Carmack, C., Ryan, M., Liu, J., Derbyshire, V., and Lawrence, C. 2001. Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. Nucleic Acids Res. 29(3): 774–782.

[McGuire *et al.* 2000] McGuire, A., Hughes, J., and Church, G. 2000. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. Genome Res. 10(6): 744–757.

[Mukherjee *et al.* 2004] Mukherjee, S., Berger, M.F., Jona, G., Wang, X.S., Muzzey, D., Snyder, M., Young, R.A., Bulyk, M.L. 2004. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. Nature Genetics 36(12): 1331–1339.

[Neuwald *et al.* 1995] Neuwald, A., Liu, J., Lawrence C. 1995. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. Protein Sci. 4(8): 1618–32.

[Osada *et al.* 2004] Osada, R., Zaslavsky, E., and Singh, M. 2004. Comparative analysis of methods for representing and searching for transcription factor binding sites. Bioinformatics 20(18): 3516–3525.

[Pavesi *et al.* 2004] Pavesi, G., Mereghetti, P., Mauri, G. and Pesole. G. 2004. Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. Nucleic Acids Res. 32: W199–W203.

[Pevzner and Sze 2000] Pevzner, P. and Sze, S. 2000. Combinatorial approaches to finding subtle signals in DNA sequences. In Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pages 269–278. AAAI Press.

[Price *et al.* 2003] Price, A., Ramabhadran, S. and Pevzner, P. 2003. Finding subtle motifs by branching from sample strings. Bioinformatics 19: 149–155.

[Reinert *et al.* 1997] Reinert, K., Lenhof, H.P., Mutzel, P., Mehlhorn, K., and Kececioglu, J. A branch-and-cut algorithm for multiple sequence alignment. In Proceedings of the First Annual International Conference on Computational Molecular Biology, pages 241–249. ACM Press.

[Rigoutsos and Floratos 1998] Rigoutsos, I., and Floratos, A. 1998. Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm. Bioinformatics 14(1): 55–67.

[Robison *et al.* 1998] Robison, K. and McGuire, A. M. and Church, G. M. 1998. A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete *Escherichia coli* K-12 Genome. J. Mol. Biol. 284: 241–254.

[Schuler *et al.* 1991] Schuler, G., Altschul, S., and Lipman, D. 1991. A workbench for multiple alignment construction and analysis. Proteins 9(3): 180–190.

[Sinha and Tompa 2003] Sinha, S. and Tompa, M. 2003. YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Res. 31(13): 3586-3588.

[Sze *et al.* 2004] Sze, S.-H., Lu, S. and Chen, J. 2004. Integrating Sample-driven and Pattern-driven Approaches in Motif Finding. Proceedings of the Fourth Workshop on Algorithms in Bioinformatics (WABI), pages 438–449.

[Tavazoie *et al.* 1999] Tavazoie, S., Hughes, J. D, Campbell, M. J., Cho, R. J., Church, G.M. 1999. Systematic determination of genetic network architecture. Nature Genetics 22(3): 281–285.

[Thompson *et al.* 2003] Thompson, W., Rouchka, E. C. and Lawrence, C. E. 2003. Gibbs Recursive Sampler: finding transcription factor binding sites, Nucleic Acids Research, 31(13): 3580-3585.

[Tompa 1999] Tompa, M. An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. In Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pages 262–71. AAAI Press, 1999.

[Tompa *et al.* 2005] Tompa, M., Li, N., Bailey, T. L., Church, G.M., De Moor, B., Eskin, E., Favorov, A. V., Frith, M.C., Fu, Y., Kent, W. J., Makeev, V. J., Mironov, A. A., Noble, W. S., Pavesi, G., Pesole, G., Regnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., and Zhu, Z. 2005. Assessing computational tools for the discovery of transcription factor binding sites. Nature Biotechnology 23(1): 137–44.

[van Helden *et al.* 2000] van Helden, J., Rios, A.F. and Collado-Vides, J. 2000. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. Nucleic Acids Res. 28(8): 1808–1818.

[Vingron and Pevzner 1995] Vingron, M., and Pevzner, P. 1995. Multiple sequence comparison and consistency on multipartite graphs. Advances in Applied Mathematics 16: 1–22.

[Workman and Stormo 2000] Workman, C.T. and Stormo, G.D. ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. In Proceedings of the Fifth Pacific Symposium on Biocomputing, pages 467–478, 2000.

[Wang and Jiang 1994] Wang, L. and Jiang, T. 1994. On the complexity of multiple sequence alignment. Journal of Computational Biology, 1: 337–348.

[Wingender *et al.* 1996] Wingender, E., Dietze, P., Karas, H., and Knppel, R. 1996. TRANSFAC: A database on transcription factors and their DNA binding sites. Nucleic Acids Res. 24: 238241.

Dataset	**# Seq.**	**Motif Len.**	$\lvert V\rvert$	**DEE Methods**	$\lvert V\rvert$ **after DEE**
Lipocalin	5	16	861	(1)	58
Helix-Turn-Helix	30	18	6871	(1,2)	145
Tumor Necrosis Factor	10	17	2329	(1)	593
Zinc Metallopeptidase	10	12	7760	(1,2,3)	10
Immunoglobulin Fold	18	14	7425	(1,2,3)	114

Table 1: Descriptions of the protein datasets. The first two datasets are from [Lawrence *et al.* 1993], the next two from [Lukashin and Rosa 1999], and the last one from [Neuwald *et al.* 1995]. ***# Seq.*** gives in the number of input protein sequences; ***Motif Len.*** gives the length of the protein motif searched for; $|V|$ gives the number of vertices in the original graph constructed from the dataset; ***DEE Methods*** gives the methods involved in pruning the graph and are denoted by (1) *clique-bounds* DEE, (2) *graph decomposition*, and (3) tighter constrained bounds. $|V|$ ***after DEE*** gives the size of the graph after graph pruning.

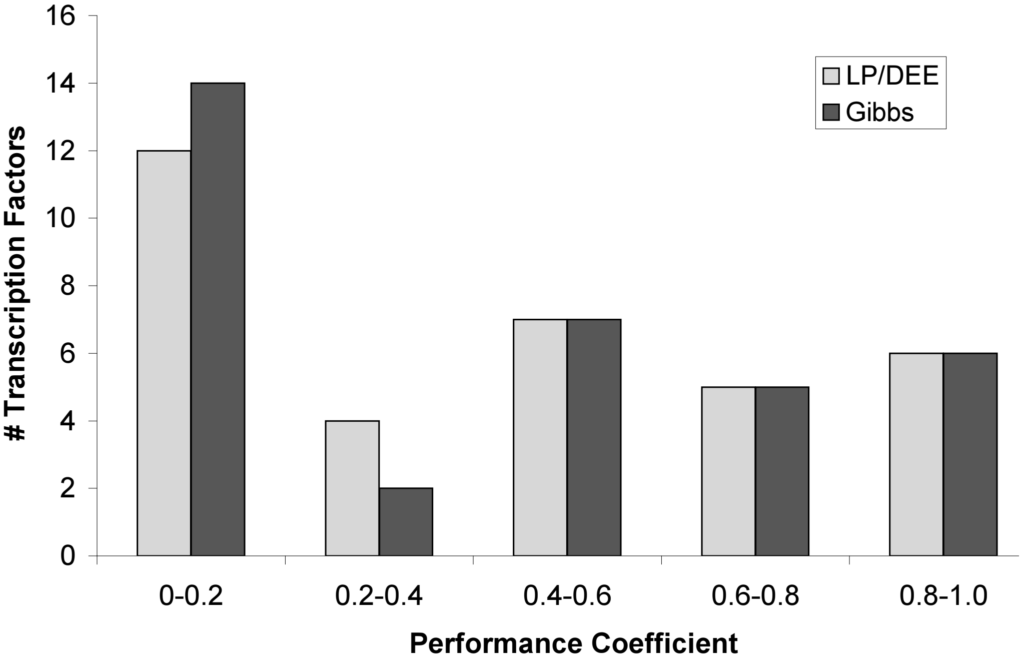

Figure 1: Degree of overlap (performance coefficient) between known regulatory sites [Robison *et al.* 1998] and motifs identified by LP/DEE and Gibbs Motif Sampler [Thompson *et al.* 2003]. For each method, the height of the bar indicates the number of transcription factor datasets, for which the performance coefficient falls in the specified interval. Higher overlap indicates better correspondence with known binding sites. *Crp* is excluded from the data as its motif was not identified by either method.

DNA region	**Species**	**Motif (id)**
Growth-hormone 5' UTR + promoter (380 bp)	Salmon, trout, white fish, seriola, lates, tilapia, fugu, grass carp, catfish, chicken, rat, mouse, dog, sheep, goat, human	TATAAAAA (7)
Histone H1 5' UTR + promoter (650 bp)	Chicken, duck, frog, mouse	AAACAAAAGT (2)
C-fos 5' UTR + promoter (800 bp)	Tetraodon, chicken, mouse, hamster, pig, human	CCATATTAGG
C-fos first intron (376 to 758 bp)	Fugu, tetraodon, chicken, pig, mouse, hamster, human	AGGGATATTT (3)
Interleukin-3 5' UTR + promoter (490 bp)	Rat, mouse, cow, sheep, human, macaca	TGGAGGTTCC (3)
C-myc second intron (971 to 1376 bp)	Chicken, pig, rat, marmoset, gibbon, human	TTTGCAGCTA (5)
C-myc 5' promoter (1000 bp)	Goldfish, frog, chicken, rat, pig, marmoset, human	GCCCCTCCCG
Insulin family 5' promoter (500 bp)	Human, chimp, aotus, pig, rat (I, II), mouse (I, II)	GCCATCTGCC (2) TAAGACTCTA (1) CTATAAAGCC (3) CAGGGAAATG (4)

Table 2: Motifs identified with use of phylogenetic information. All datasets tested are from [Blanchette and Tompa 2002]. ***DNA region*** details the DNA regions considered; ***Species*** lists the species and isoforms considered; ***Motif (id)*** identifies the consensus sequence of the discovered motif and its correspondence with the motifs of [Blanchette and Tompa 2002] where applicable. All listed motifs have been documented as regulatory elements in TRANSFAC [Wingender *et al.* 1996]. For datasets other than the *insulin* dataset only the best motif is reported, and for the *insulin* dataset multiple motifs are reported in order of discovery.