

# From Profiles to Patterns and Back Again: A Branch and Bound Algorithm for Finding Near Optimal Motif Profiles

Eleazar Eskin

Department of Computer Science and Engineering  
University of California, San Diego

eeskin@cs.ucsd.edu

## ABSTRACT

An important part of deciphering gene regulatory mechanisms is discovering transcription factor binding sites. In many cases, these sites can be detected because they are often overrepresented in genomic sequences. The detection of the overrepresented signals in sequences, or *motif-finding* has become a central problem in computational biology. There are two major computational frameworks for attacking the motif finding problem which differ in their representation of the signals. The most popular is the profile or PSSM (Position Specific Scoring Matrix) representation. The goal of these algorithms is to obtain probabilistic representations of the overrepresented signals. Another is the consensus pattern or pattern with mismatches representation which represents a signal as discrete consensus pattern and allows some mismatches to occur in each instance of the pattern. The advantage of profiles is the expressiveness of their representation while the advantage of the consensus pattern approach is the existence of efficient algorithms that guarantee discovery of the best patterns. In this paper we present a unified framework for motif finding which encompasses both the profile representation and the consensus pattern representation. We prove that the problem of discovering the best profiles can be solved by considering a degenerate version of the problem of finding the best consensus patterns. The main advantage of our framework is that it motivates a novel algorithm, MITRA-PSSM, which discovers profiles, yet provides some of the guarantees of discovering the best signals. The algorithm searches for best profiles with respect to information content which is the same criterion of popular algorithms such as MEME and CONSENSUS. MITRA-PSSM is specifically designed for searching for profiles in this framework and introduces a novel notion of scoring consensus patterns, *discrete information content*. MITRA-PSSM is available for public use via webserver at <http://www.calit2.net/compbio/mitra/>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RECOMB'04, March 27–31, 2004, San Diego, California, USA.

Copyright 2004 ACM 1-58113-755-9/04/0003 ...\$5.00.

## Categories and Subject Descriptors

J.3 [Life and Medical Sciences]: Biology and Genetics

## General Terms

Algorithms

## Keywords

motif-finding, transcription factor binding sites, profiles, patterns

## 1. INTRODUCTION

With the availability of powerful methods for prediction of genes in genomic sequences, many of the genes in sequenced genomes have been identified. The next major challenge is discovering how these genes interact with and regulate each other. An important part of deciphering gene regulatory mechanisms is discovering transcription factor binding sites. In many cases, these sites can be detected because they are often overrepresented in genomic sequences. The detection of the overrepresented signals in sequences, or *motif-finding* has become a central problem in computational biology over the last 20 years.

Most approaches over the last two decades can be split roughly into two categories corresponding to two major computational frameworks for attacking the motif finding problem differing in their representation of the motifs. The most widely used is the profile or PSSM (Position Specific Scoring Matrix) representation. The goal of these algorithms is to obtain a generative probabilistic representation of the overrepresented signals. These algorithms view the representation of the signals as continuous. Each position in a profile consists of a vector of probabilities for observing each nucleotide at that position. The algorithms in this framework attempt to discover profiles that maximize a criterion such as the information content score. Popular approaches within this framework include MEME[1], CONSENSUS[10], Gibbs sampler[13, 14], and AlignACE[11]. The major advantages of this framework are that profiles are very powerful representation of the signals and the scoring function is motivated by an underlying probabilistic model. Additional information in the motif search such as background statistics, expression data, aligned genomes, functional categories of genes, and position information can easily be incorporated in the context of the probabilistic model as shown below. A major disadvantage is that the problem of discovering the

“best” profile is inherently very difficult. For a signal of length  $l$ , the problem reduces to finding the global maximum of a non-convex continuous  $3l$  dimensional function. Most algorithms in this framework can be characterized as performing stochastic optimization or greedy searches and give no guarantees of finding the best profile or even profiles close to the best profiles.

The second framework uses the consensus pattern or patterns with mismatches representation. These algorithms define a signal to be a consensus pattern and allow up to a certain amount of mismatches to occur in each instance of the pattern. The goal of these algorithms is to recover the consensus pattern with the highest number of instances. Recent approaches within this framework include random projections[4], combinatorial based approaches[15], and MULTIPROFILER[12]. A variant of this formulation determines the best consensus patterns based on some measure of statistical significance of the number of instances. The major advantage of many of these algorithms is that because of the discrete nature of the patterns, even the classical algorithms such as the 1984 Waterman et al.[20, 9] algorithm guaranteed to find the best scoring patterns. Recently, more efficient branch and bound algorithms for finding consensus patterns such as the algorithms of Sagot [16] and MITRA[8] have been presented as well as more sophisticated methods for determining the statistical significance of patterns[6]. In addition, some consensus pattern algorithms such as MITRA[7] are modified to guarantee to find the highest scoring patterns for a given scoring function. However, a major disadvantage to the consensus pattern or pattern with mismatches framework is the representation of the signals themselves. In general, the profile representation is much more expressive than the consensus pattern representation. For example, consensus patterns in general can not distinguish between conserved and unconserved positions. In addition, the traditional formulation of consensus patterns can not represent positions where multiple bases can occur without increasing the alphabet to include IUPAC symbols. Even using the IUPAC alphabet[18], the patterns can not represent different degrees of conservation among different bases.

The main difference between these two frameworks can be summarized by noting that the continuous profile representation is more expressive at the cost of the guaranteed convergence properties of the less expressive discrete consensus pattern representation. There is a long standing debate about which framework performs better[3, 19]. Ideally, the best approach would combine the expressiveness of the profile representation with the guarantees of convergence of the consensus pattern approach.

In this paper, we present a novel algorithm, MITRA-PSSM, for discovering profiles for overrepresented signals in sequences which provides some of the guarantees of discovering the best signals similar to the consensus pattern approaches. The key to our approach is formulating a common framework for pattern discovery which encompasses both the profile and consensus pattern representations. In this framework, we introduce a new notion of scoring consensus patterns, *discrete information content*, which views the consensus pattern as a discrete profile and the space of consensus patterns as a discretization of the space of profiles. Using this common framework, we are able to define algorithms that function on consensus patterns while performing an analogous search over the space of profiles.

MITRA-PSSM decomposes the space of profiles into disjoint subspaces each which is associated with a consensus pattern that is a discrete approximation of the profile subspace. The consensus patterns “cover” their respective profile subspaces which means that any instance of a profile in the subspace is also an instance of the consensus pattern. The MITRA-PSSM algorithm consists of two main phases. The first phase performs a branch and bound search to identify promising profile subspaces. This is done by eliminating the subspaces of the space of all profiles which cannot score higher than the minimum motif strength threshold. The search of the pattern space of MITRA-PSSM is very similar to the SPELLER algorithm presented in Sagot, 1998[16]. The second part performs a local search over the remaining profile subspaces. At each step in the branch and bound search, based on the instances of the consensus pattern, we can obtain an upper bound on the score of any profile in the subspace. If this maximum score is below our minimum motif strength threshold we can eliminate all of the profiles from this subspace from consideration.

The advantage of the MITRA-PSSM algorithm as opposed to other algorithms is that we can obtain some guarantees about the profiles that the algorithm finds. The algorithm narrows down the location of the best PSSMs to a few subspaces. Although it is not able to identify exactly where in these subspaces the best PSSMs lie, by considering the width of the subspace the algorithm gives an upper bound to the distance of the discovered PSSMs to the highest scoring PSSMs.

Unlike many profile discovery algorithms, MITRA-PSSM is deterministic and can be used to find multiple high scoring profiles in a single run. This eliminates the need to mask high scoring signals to discover the remaining signals as in algorithms such as MEME.

We also introduce a very general probabilistic formulation of information content which allows us to incorporate a wide range of information into the motif search. This includes background statistics, aligned genomes, gene functional categories, gene expression data, known transcription factors, and positional information which can all be introduced into the motif search using a variant of the problem formulation. MITRA-PSSM can perform the motif search in this generalized setting.

We demonstrate our algorithms on discovering PSSMs in the complete set of promoters of *E. Coli*, a very large samples of close to 250,000 bases where we know there are many different signals.

The paper is organized as follows. In Section 2 we introduce the probabilistic model and formulate the motif finding problem. In Section 3 we introduce the notion of discrete information content. In Section 4 we introduce the notion of “cover” and relate profile subspaces to discrete information content approximations. In Section 5 we derive an upper bound on the score of a profile given a superset of its instances. In Section 6 we present MITRA-PSSM. In Section 7 we demonstrate the applicability of MITRA-PSSM to biological data. Finally, in Section 8 we conclude and discuss future directions.

## 2. PSSMS, INFORMATION CONTENT AND MAXIMUM LIKELIHOOD

Profiles and the information content score are best un-

derstood in the context of a generative probabilistic model. Here we derive the information content score from a generative model to motivate the formulation of the motif finding optimization problem. The probabilistic model defines how the data was generated. For simplicity of the formulation, we initially assume that we are given a single long sequence as our sample and we extend to samples consisting of multiple sequences in Section 6.4. Given a sample  $D$  with  $m$  substrings of length  $l$ , we denote the  $j$ th length  $l$  substring of the sample by  $x_j$ , and the  $i$ th position of the  $j$ th substring as  $x_j^i$ .

For every substring in the data, we assume that it was generated in one of two possible ways. Either the substring is an instance of the signal and generated by the profile, or otherwise it is generated by the background model[17]. The probabilistic model can be viewed as a mixture model with two components each corresponding to how the instance was generated.

A profile  $S$  is represented by a  $4 \times l$  matrix providing a probabilistic interpretation of the signal. For the alphabet  $\Sigma = \{A, C, G, T\}$ , let the entries at the  $i$ th position in a profile,  $S$ , be denoted  $\{s_A^i, s_C^i, s_G^i, s_T^i\}$  respectively. Since these entries represent probabilities,  $\sum_{\sigma \in \Sigma} s_\sigma^i = 1$ . We denote the background probabilities  $\{b_A, b_C, b_G, b_T\}$ .

For any substring  $x_j$ , the probability of the substring being generated by the profile  $S$  is  $S(x_j) = \prod_{i=1}^l s_{x_j^i}^i$  and the probability of  $x_j$  being generated by the background sample is  $B(x_j) = \prod_{i=1}^l b_{x_j^i}$ . We denote a set of hidden variables  $g_1, \dots, g_j, \dots, g_m$ , one for each substring such that  $g_j = 1$  if  $x_j$  is generated by the profile  $S$  and  $g_j = 0$  otherwise. We define the prior probability of an instance  $x_j$  being generated by the sample as  $P_j(S)$ . Assuming this model, given the value of these hidden variables, the likelihood of the sample is

$$\begin{aligned} L &= \prod_{x_j: g_j=1} P_j(S) S(x_j) \prod_{x_j: g_j=0} (1 - P_j(S)) B(x_j) \\ &= \prod_{x_j: g_j=1} \frac{P_j(S)}{(1 - P_j(S))} \frac{S(x_j)}{B(x_j)} \prod_{x_j} (1 - P_j(S)) B(x_j) \quad (1) \end{aligned}$$

This likelihood will be maximized if  $g_j = 1$  when  $P_j(S) S(x_j) > (1 - P_j(S)) B(x_j)$  which is equivalent to

$$g_j = \begin{cases} 1 & \text{if } \frac{S(x_j)}{B(x_j)} > \frac{(1 - P_j(S))}{P_j(S)} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that the last term of the product in equation (1) is a constant that depends only on the sample and the background probability which we denote  $P(B) = \prod_{x_j} (1 - P_j(S)) B(x_j)$ . We can write the maximum likelihood as

$$L = \prod_{x_j: \frac{S(x_j)}{B(x_j)} > \frac{(1 - P_j(S))}{P_j(S)}} \frac{P_j(S)}{(1 - P_j(S))} \frac{S(x_j)}{B(x_j)} P(B) .$$

If we assume the prior probability for any instance being part of the signal is the same, i.e.,  $P(S) = P_j(S)$  for all  $j$  and if we consider the maximum log likelihood<sup>1</sup> then

$$\log(L) = \sum_{x_j: \log\left(\frac{S(x_j)}{B(x_j)}\right) > t} \left( \log\left(\frac{S(x_j)}{B(x_j)}\right) - t \right) + \log(P(B))$$

<sup>1</sup>Throughout this paper we use log base 2 because information content is typically measured in bits.

where  $t = \log\left(\frac{(1 - P(S))}{P(S)}\right)$ . A substring  $x_j$  is an instance of the profile if it has a score above the threshold  $t$ . If we drop the term  $\log(P(B))$  since it is not dependent on  $S$ , we can define the information content for a profile  $IC(S)$  as

$$IC(S) = \sum_{x_j: \log\left(\frac{S(x_j)}{B(x_j)}\right) > t} \left( \log\left(\frac{S(x_j)}{B(x_j)}\right) - t \right) . \quad (3)$$

Note that maximizing  $IC(S)$  is equivalent to finding the profile that maximizes the likelihood under the probabilistic model.

This version of the information content score is very similar to the traditional formulation where the information content score is computed over a collection of instances. Given a collection  $C$  of  $N$  substrings, let  $S$  be the profile generated from computing the consensus profile of the  $N$  substrings. We denote  $n_\sigma^i = |\{x_j : x_j \in C, x_j^i = \sigma\}|$  the number of times  $\sigma$  occurs in position  $i$  in the collection. The classical information content score for this collection  $IC_C(C)$  is written

$$IC_C(C) = \sum_{1 \leq i \leq l} \sum_{\sigma \in \{A, C, G, T\}} \frac{n_\sigma^i}{N} \log\left(\frac{n_\sigma^i}{b_\sigma^i N}\right) .$$

The classical information content is the Kullback-Leibler information or relative entropy[10] and is a measure of the ‘‘strength’’ of the profile or distance of the profile from the background distribution. As compared to our formulation, the classical information content assumes that  $P(S) = 1 - P(S)$  which implies  $t = 0$ . Note that  $IC(S) = N(IC_C(C) - t)$  if  $s_\sigma^i = \frac{n_\sigma^i}{N}$  which is the case at the maximum likelihood as shown in Lemma 3. We denote the profile associated with a collection  $C$  as  $S_C$ . The main difference between our notion of information content and the classical notion is that we multiply the relative entropy by the number of instances of the signal. However, in the Hertz and Stormo, 1999[10] technique for computing the  $p$ -value of a profile, uses as the statistic the product of the number of instances and the classical information content. Understandably, the  $p$ -value of a motif will depend *both* on the strength of the profile and the number of instances of the motif. Thus the statistical significance of motifs under both notions of information content are equivalent even though the scores are defined differently.

A useful representation for profiles is a weight matrix. A signal of length  $l$  is defined by a  $4 \times l$  weight matrix  $W$  such that the  $i$ th column of the matrix denoted  $\{w_A^i, w_C^i, w_G^i, w_T^i\}$  represents the  $i$ th position of a signal. We can define the weight matrix  $W_C$  for a collection  $C$  with a weight matrix  $w_\sigma^i = \log\left(\frac{n_\sigma^i}{b_\sigma^i N}\right)$ . For a profile  $S$ , we define a weight matrix  $W_S$  derived from the profile such that  $w_\sigma^i = \log\left(\frac{s_\sigma^i}{b_\sigma^i}\right) - t/l$ .

Given a substring  $x_j$  and a weight matrix  $W$ , the score of the substring with respect to  $W$  is defined to be  $score(W, x_j) = \sum_{1 \leq i \leq l} w_{x_j^i}^i$ . The score for the signal  $W$ ,  $score(W, D)$  over the data set is simply the sum of the scores of the positive scoring substrings,  $\sum_{j: score(W, x_j) > 0} score(W, x_j)$ . We denote the instances of the signal  $W$ ,  $I(W, D) = \{x_j : score(W, x_j) > 0\}$ , as the subset of the data which has a positive score for the weight matrix. Note that the instances of a signal  $W$ ,  $I(W, D)$  are completely determined by the signal and the data. We define the neighborhood of a signal  $N(W)$  as the set of substrings of length  $l$  that have positive

score for the signal,  $N(W) = \{x \in \Sigma^l : score(W, x) > 0\}$ .

From a profile  $S$  and its corresponding weight matrix  $W_S$ , we can compute its instances  $I(W_S, D)$  and information content  $IC(S)$ . We can formulate the motif finding problem as a search over profiles.

**The Motif Finding Problem (Profiles).** *Given a sample  $D$  consisting of substrings  $x_j$ , determine the profile  $S$  such that  $IC(S)$  is maximized.*  $\square$

Similarly, from a collection of instances  $C$ , we can compute its profile  $S_C$ , and its corresponding information content. This represents the dual nature of the motif finding problem. We can formulate the problem of motif finding as a search over subsets of the substrings in the data.

**The Motif Finding Problem (Instances).** *Given a sample  $D$  consisting of substrings  $x_j$ , choose a collection of instances  $C \subset D$  such that  $IC(S_C)$  is maximized.*  $\square$

Each formulation of the problem has an equivalent solution. The dual nature of the problem naturally leads to the use of the EM algorithm[5] such as in MEME[1]

## 2.1 Relation to Traditional Profile Algorithms

The probabilistic framework we present is closely related to many of traditional profile motif finding algorithms. The closest is the CONSENSUS[10] objective function which is equivalent to our formulation given that  $t = 0$ . However, one of the algorithms they present introduces a notion of “crude information content” which is intuitively very similar to using a threshold. The EM[1] and Gibbs Sampling[13, 14, 11] methods differ from our framework in that they assume Bayesian priors for estimating the profile from the data and compute the *maximum a posteriori* (MAP) likelihood instead of the maximum likelihood. The priors affect the estimation of the profile by incorporate pseudo-counts. Extending this approach to incorporate priors and MAP estimates are directions for future research. Recently, a class of discriminative scoring approaches to motif finding have been presented[2, 17] which are inherently different from the model described here. These scoring approaches attempt to find motifs which discriminate between a set of sequences of interest from a background set of sequences.

## 2.2 Incorporating Additional Information into the Motif Search

We can incorporate additional information related to the motif search into the probabilistic model by dropping the assumption that  $P(S) = P_j(S)$  for all  $j$ . Instead, we assign each  $x_j$  a different value of  $P_j(S)$  to take into account the additional information. In this case, we have a different threshold for each instance  $t_j = \log\left(\frac{(1-P_j(S))}{P_j(S)}\right)$  and the information content for a profile is

$$IC(S) = \sum_{x_j : \log\left(\frac{S(x_j)}{B(x_j)}\right) > t_j} \left(\log\left(\frac{S(x_j)}{B(x_j)}\right) - t_j\right). \quad (4)$$

Using this formalism, we can incorporate many types of information into the search. For example, if we want to consider a background model that assumes an  $k$  order Markov chain instead of the uniform background distribution as in Equation 3, we can set the threshold  $t_j = t + \log(B_k(x_j)) - \log(B(x_j))$  where  $B_k(x_j)$  is the probability of observing  $x_j$  under the  $k$  order Markov distribution and  $t$  is the original global threshold. In this framework, the threshold is

Symbol	Meaning	Origin of Description
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
M	A or C	aMino
R	A or G	puRine
W	A or T	Weak interaction (2 H bonds)
S	C or G	String interaction (3 H bonds)
Y	C or T	pYrimidine
K	G or T	Keto
V	A or C or G	not-T (not-U), V follows U
H	A or C or T	not-G, H follows G
D	A or G or T	not-C, D follows C
B	C or G or T	not-A, B follows A
N	A or C or G or T	aNy

**Table 1: IUPAC Alphabet for Nucleotide Sequences**

adjusted for each substring  $x_j$ . Intuitively, for sequences that are overrepresented in the  $k$  order Markov distribution relative to a 0 order Markov distribution (such as poly-A signals) the difference  $\log(B_k(x_j)) - \log(B(x_j))$  will be positive and the threshold will increase making it harder for those substrings to become part of the motif. On the other hand, for high complexity signals, the difference will be negative and the threshold will decrease making it easier for those substrings to be part of the motif.

This flexible mechanism allows us to incorporate many different types of information into the search simply by setting the value of  $P_j(S)$  appropriately which implicitly sets a per-substring threshold  $t_j$ . The types of information that can be incorporated are aligned genomes, expression data, gene functional categories, known transcription factors, and positional information. For example, if we are looking for motifs in a sample where we have an aligned genome to the sample, we can set  $P_j(S)$  to a high value if the substring  $x_j$  is conserved in the alignment and to a low value otherwise. If we are looking for motifs in a set of co-expressed genes or present in a set of genes in the same functional category, we can set  $P_j(S)$  to a high value for any substring in one of these genes and  $P_j(S)$  to a low value for any other substring. We can also set  $P_j(S)$  based on the proximity of substring  $x_j$  to known transcription factors or proximity to a certain position. Exactly how the values for  $P_j(S)$  are set depends on the assumptions of the probabilistic model for incorporating the additional information.

The problem formulation for finding the best motifs while taking into account this information changes slightly. Instead of a single threshold for all substrings, the threshold varies per substring as in Equation (4).

## 3. DISCRETE INFORMATION CONTENT

The consensus pattern representation has inherent difficulties in expressing profile type motifs. In general, consensus patterns can not take into account different degrees of conservation in positions and differing background distributions. Consensus patterns can be made significantly more expressive if we incorporate the IUPAC alphabet shown in Table 1. IUPAC symbols provide the ability to express degenerate patterns or patterns where in one position there is more than one possible symbol. However, if we allow for degenerate symbols in the consensus patterns, it is not clear how to define differing penalties for mismatches in degenerate positions.

In this section, we present a new type of scoring for con-

		Positions					
		1	2	3	4	5	6
(a)	<b>A</b>	1.00	0.08	0.00	0.75	0.10	0.00
	<b>C</b>	0.00	0.75	0.00	0.08	0.40	1.00
	<b>G</b>	0.00	0.08	0.50	0.08	0.10	0.00
	<b>T</b>	0.00	0.08	0.50	0.08	0.40	0.00

		Positions					
		1	2	3	4	5	6
(b)	<b>A</b>	2	-1.58	$-\infty$	1.58	-1.32	-1.32
	<b>C</b>	$-\infty$	1.58	$-\infty$	-1.58	.69	-1.32
	<b>G</b>	$-\infty$	-1.58	1	-1.58	-1.32	.69
	<b>T</b>	$-\infty$	-1.58	1	-1.58	.69	.69

**Figure 1: Profile (a) and Weight Matrix (b) induced by discrete information content pattern *AcKayC*.**

sensus patterns, *discrete information content*, which still preserves the discrete nature of consensus patterns, yet provides a score which is motivated by information content. The basic idea is to assign to each symbol in our discrete information content alphabet,  $\Sigma_{DIC}$ , a canonical one position profile. Each instance is scored using the weight matrix generated by the profiles along the positions of the pattern. Discrete information content is very flexible. We can introduce different symbols into the alphabet to represent different levels of conservation. An example of a discrete information content alphabet is shown in Figure 2 with the weights for each symbol computed assuming a uniform background distribution.

We compute the instances of a discrete information content pattern using the weight matrix associated with the pattern. For example, consider a pattern *AcKayC* (shown in Figure 1) and instances *ACGACC* and *ACCACC*. The first instance has score  $2 + 1.58 + 1 + 1.58 + .69 + 2 = 8.85$ . The second instance has a score of  $-\infty$  since the third position is a *C* while the symbol *K* only matches bases *G* and *T*.

Using the results of the next two sections, we will show that finding the best profiles corresponds to finding the best patterns under an appropriate discrete information content alphabet and we present MITRA-PSSM, an algorithm for performing this search.

## 4. COVERING PROFILE SUBSPACES WITH CONSENSUS PATTERNS

We can analyze relationships between profiles and their discrete information content approximation based on their sets of instances. If  $N(W) \subseteq N(W')$ , then we say that signal  $W'$  covers signal  $W$  and denote this relation  $W \subseteq W'$ . This implies both  $|N(W)| \leq |N(W')|$  and  $I(W, D) \subseteq I(W', D)$ .

The notion of covering is the key to our analysis. Our goal will be to split the space of profiles into disjoint subspaces and cover each profile with a discrete information content pattern. In our case,  $W$  will be a weight matrix associated with any profile in a subspace and  $W'$  will be a weight matrix associate with a discrete information content pattern. We will want that  $W \subseteq W'$  which will insure that the instance of the pattern are the superset of the instances of the profile. Using this pattern, we can then recover the instances for this subspace. In the following section, we will show how we can obtain an upper bound on the information content

of the highest scoring profile in the subspace from this set of instances.

For our analysis, given two weight matrices  $W$  and  $W'$ , we are interested in determining in general whether or not  $W \subseteq W'$ . We make use of the following Lemma which follows directly from the definition of  $W \subseteq W'$ .

LEMMA 1.  $W \subseteq W'$  if and only if  $N(W) \cap (\Sigma^l - N(W')) = \emptyset$ .

The condition of Lemma 1 can be checked to verify the relation between two weight matrices.

### 4.1 Partition of the Space of Profiles

For the purposes of our analysis, we consider the relationships between signals based on profiles and signals based on discrete information content patterns. Given a consensus pattern  $P$ , consider the profile  $S$  with a threshold  $t$  such,  $W_S \subseteq W_P$ . In this case we know that the instances of  $W_P$  are a superset of the instances of the profile  $S$ . The score of the profile  $S$ , depends only on the instances of  $P$ . If we can determine that these instances can not score above our minimum score threshold, we can determine that the profile can not score above the threshold.

We are interested in disjointly partitioning the complete space of profiles for a given threshold  $t$  such that the partition has the following property. Each partition  $\mathcal{S}$  is assigned a unique consensus pattern  $P$  and for every profile  $S \in \mathcal{S}$  in the subspace,  $W_S \subseteq W_P$ . If this is the case we can then consider only the instances of the pattern  $P$  when searching locally within the subspace  $\mathcal{S}$ . Furthermore, we can obtain an upper bound on the score of all profiles in  $\mathcal{S}$  based on the instances of  $P$  since they are a superset of any of the profiles in the subspace as shown below. If the upper bound is below our minimum score threshold, we can rule out the entire subspace of profiles  $\mathcal{S}$ .

Since discrete information content is a discrete approximation for a profile, we introduce an approximation constant  $\Delta t$  which compensates for this approximation at each position.  $\Delta t$  represents the difference in the score of a substring in a position in the discrete information content and the score of a substring in a profile at that position. Effectively,  $\Delta t$  reduces the signal threshold to  $t - l\Delta t$ . A substring is considered an instance of the discrete information content pattern if its score is at least  $t - l\Delta t$ . If an instance score is above the threshold, the instance contributes the difference of the instance score and the threshold to the overall signal score. For any discrete information content pattern  $P$  we construct a weight matrix  $W_P$  by subtracting  $\frac{t}{l} - \Delta t$  from each entry of the weight profile corresponding to the discrete information content pattern.

Using discrete information content, we can obtain a condition on partitioning profiles into subspaces.

LEMMA 2. Consider a subspace of profiles  $\mathcal{S}$  with threshold  $t$  and a pattern  $P$  with approximation constant  $\Delta t$ .  $W_S$  is the weight matrix derived from any profile  $S \in \mathcal{S}$  with threshold  $t$  and  $W_P$  is the weight matrix derived from the discrete information content pattern  $P$  with threshold  $t - l\Delta t$ . For all  $S \in \mathcal{S}$   $W_S \subseteq W_P$  if for all  $\sigma$ ,  $1 \leq i \leq l$ ,

$$w_{S\sigma}^i - \Delta t < w_{P\sigma}^i \quad (5)$$

We point out that Lemma 2 is the condition necessary for partitioning profiles into subspaces and not a method for

Symbol	A	C	G	T	N	
Profile	A	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} .25 \\ .25 \\ .25 \\ .25 \end{pmatrix}$
	C					
	G					
	T					
Weight	A	$\begin{pmatrix} 2 \\ -\infty \\ -\infty \\ -\infty \end{pmatrix}$	$\begin{pmatrix} -\infty \\ 2 \\ -\infty \\ -\infty \end{pmatrix}$	$\begin{pmatrix} -\infty \\ -\infty \\ 2 \\ -\infty \end{pmatrix}$	$\begin{pmatrix} -\infty \\ -\infty \\ -\infty \\ 2 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$
	C					
	G					
	T					

  

Symbol	a	c	g	t	
Profile	A	$\begin{pmatrix} .75 \\ .08 \\ .08 \\ .08 \end{pmatrix}$	$\begin{pmatrix} .08 \\ .75 \\ .08 \\ .08 \end{pmatrix}$	$\begin{pmatrix} .08 \\ .08 \\ .75 \\ .08 \end{pmatrix}$	$\begin{pmatrix} .08 \\ .08 \\ .08 \\ .75 \end{pmatrix}$
	C				
	G				
	T				
Weight	A	$\begin{pmatrix} 1.58 \\ -1.58 \\ -1.58 \\ -1.58 \end{pmatrix}$	$\begin{pmatrix} -1.58 \\ 1.58 \\ -1.58 \\ -1.58 \end{pmatrix}$	$\begin{pmatrix} -1.58 \\ -1.58 \\ 1.58 \\ -1.58 \end{pmatrix}$	$\begin{pmatrix} -1.58 \\ -1.58 \\ -1.58 \\ 1.58 \end{pmatrix}$
	C				
	G				
	T				

  

Symbol	M	R	W	S	Y	K	
Profile	A	$\begin{pmatrix} .5 \\ .5 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} .5 \\ 0 \\ .5 \\ 0 \end{pmatrix}$	$\begin{pmatrix} .5 \\ 0 \\ 0 \\ .5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ .5 \\ 0 \\ .5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ .5 \\ 0 \\ .5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ .5 \\ .5 \end{pmatrix}$
	C						
	G						
	T						
Weight	A	$\begin{pmatrix} 1 \\ 1 \\ -\infty \\ -\infty \end{pmatrix}$	$\begin{pmatrix} 1 \\ -\infty \\ 1 \\ -\infty \end{pmatrix}$	$\begin{pmatrix} 1 \\ -\infty \\ -\infty \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \\ -\infty \\ -\infty \end{pmatrix}$	$\begin{pmatrix} -\infty \\ 1 \\ -\infty \\ 1 \end{pmatrix}$	$\begin{pmatrix} -\infty \\ -\infty \\ 1 \\ 1 \end{pmatrix}$
	C						
	G						
	T						

  

Symbol	m	r	w	s	y	k	
Profile	A	$\begin{pmatrix} .4 \\ .4 \\ .1 \\ .1 \end{pmatrix}$	$\begin{pmatrix} .4 \\ .1 \\ .4 \\ .1 \end{pmatrix}$	$\begin{pmatrix} .4 \\ .1 \\ .1 \\ .4 \end{pmatrix}$	$\begin{pmatrix} .1 \\ .4 \\ .4 \\ .1 \end{pmatrix}$	$\begin{pmatrix} .1 \\ .4 \\ .1 \\ .4 \end{pmatrix}$	$\begin{pmatrix} .1 \\ .1 \\ .4 \\ .4 \end{pmatrix}$
	C						
	G						
	T						
Weight	A	$\begin{pmatrix} .69 \\ .69 \\ -1.32 \\ -1.32 \end{pmatrix}$	$\begin{pmatrix} .69 \\ -1.32 \\ .69 \\ -1.32 \end{pmatrix}$	$\begin{pmatrix} .69 \\ -1.32 \\ -1.32 \\ .69 \end{pmatrix}$	$\begin{pmatrix} -1.32 \\ .69 \\ .69 \\ -1.32 \end{pmatrix}$	$\begin{pmatrix} -1.32 \\ -1.32 \\ .69 \\ .69 \end{pmatrix}$	$\begin{pmatrix} -1.32 \\ .69 \\ .69 \\ .69 \end{pmatrix}$
	C						
	G						
	T						

Figure 2: A simple discrete information content alphabet containing 21 symbols. For each symbol, the canonical 1 position profile and its corresponding weight matrix is shown. For each base, there are two symbols, the capital letter ( $A$ ) represents a completely conserved position while the lowercase level ( $a$ ) represents a position which tolerates some mismatches. For each IUPAC symbol for degenerate positions of 2 bases, the capital letter ( $K$ ) represents a position that only allows the 2 bases while a lowercase letter represents ( $k$ ) represents a position that prefers the two bases but allows the other bases. The  $N$  symbol represents unconserved positions. The weight matrices for each symbol are computed assuming a uniform background distribution.

constructing such a partition. Given any partition,  $\Delta t$  can be set large enough to satisfy this condition. For example, the partition consisting of the symbols from Figure 2 with  $\Delta t = .58$  partitions the complete space of profiles.

Intuitively, partitions with many symbols, will have small values of  $\Delta t$  to satisfy the condition. However, partitions with many symbols increase the number of PSSM subspaces and the size of the branch and bound search. A direction for future research is to design partitions with low  $\Delta t$  which have few symbols.

## 5. MAXIMUM SCORE OF A PROFILE

Given a subspace of profiles  $\mathcal{S}$  and a pattern  $P$  such that for all  $S \in \mathcal{S}$ ,  $W_S \subseteq W_P$ , we know that the instances of the pattern  $I(W_P, D)$  are a superset of the instances of the profile  $I(S, D)$ . This implies that  $score(S, D) = score(S, I(W_P, D))$ . Using this fact, we can derive an upper bound on the score of the best scoring profile  $S \in \mathcal{S}$ ,  $score(S, D)$ . The maximum score will be achieved for some profile  $S^*$  and subset  $I' \subseteq I(W_P, D)$  where each instance in the subset is positive scoring with respect to  $S^*$ . We denote  $|I'| = N'$ .

We can derive an upper bound on the score of the profile  $S^*$  by considering each position independently. We first consider the case where the threshold  $t$  is the same for every substring and later extend this to the case where there is a different threshold  $t_j$  for every substring  $x_j$ . At each position, we choose a subset of the instances and the profile entries for the position to maximize our score. The best scoring profile will score lower than this construction because we are not constrained to pick the same subset of instances at each position. We denote the set of nucleotides at position  $i$

as  $I'^i$ . Note that the size of  $I'^i$  is constant for each position  $|I'^i| = N'$ .

Out of a set of  $|I|$  nucleotides we pick a subset  $I'$ . Without loss of generality, we denote the counts of the nucleotides in  $I^i$  as  $N_A^i, N_C^i, N_G^i, N_T^i$ , counts in the subset  $I'^i$  as  $n_A^i, n_C^i, n_G^i, n_T^i$  and background probabilities  $b_A, b_C, b_G, b_T$  where  $\sum_{\sigma \in \Sigma} n_\sigma^i = |I'^i| = N'$ ,  $n_\sigma^i \leq N_\sigma^i$ . We need to pick  $n_\sigma^i$  and  $s_\sigma^i$  to maximize the score

$$\sum_{i=1}^l \sum_{\sigma \in \Sigma} n_\sigma^i \log\left(\frac{s_\sigma^i}{b_\sigma}\right) - tN' \quad (6)$$

with constraint  $\sum_{\sigma \in \Sigma} s_\sigma^i = 1$  and  $\sum_{\sigma \in \Sigma} n_\sigma^i = N'$ .

LEMMA 3. For any choice of  $n_\sigma^i$ , Equation (6) is maximized if  $s_\sigma^i = \frac{n_\sigma^i}{\sum_{\sigma \in \Sigma} n_\sigma^i}$ .

PROOF. Since  $n_\sigma^i$  have been chosen in advance, we only need to choose values of  $s_j^i$  to maximize Equation (6). We only need to consider the first part of the equation since the second is constant given  $n_\sigma^i$ . We can show this by using Lagrange multipliers with a single constraint  $\sum_{\sigma \in \Sigma} s_\sigma^i = 1$ . Using Lagrange multipliers we have

$$\begin{aligned} L(s_A^i, s_C^i, s_G^i, s_T^i, \lambda) &= \sum_{\sigma \in \Sigma} n_\sigma^i \log\left(\frac{s_\sigma^i}{b_\sigma}\right) \\ &\quad + \lambda \left( \sum_{\sigma \in \Sigma} s_\sigma^i - 1 \right) \\ \frac{\partial L(s_A^i, s_C^i, s_G^i, s_T^i, \lambda)}{\partial s_\sigma^i} &= \frac{n_\sigma^i}{s_\sigma^i \log 2} + \lambda = 0 \quad (7) \\ \frac{\partial L(s_A^i, s_C^i, s_G^i, s_T^i, \lambda)}{\partial \lambda} &= \sum_{\sigma \in \Sigma} s_\sigma^i - 1 = 0 \end{aligned}$$

From Equation (7)  $s_\sigma^i = s_{\sigma'}^i \frac{n_\sigma^i}{n_{\sigma'}^i}$ . Plugging this into the constraint we get for  $s_A^i = 1 - \sum_{\sigma \in \{C, G, T\}} s_A^i \frac{n_\sigma^i}{n_A^i}$  which simplifies to the form of Lemma 3.  $\square$

Since we now have  $s_\sigma^i$  in terms of  $n_\sigma^i$ , our maximization problem reduces to finding the set  $n_\sigma^i$  that maximizes

$$\sum_{i=1}^l \sum_{\sigma \in \Sigma} n_\sigma^i \log \left( \frac{n_\sigma^i}{b_\sigma \sum_{\sigma' \in \Sigma} n_{\sigma'}^i} \right) - tN' \quad (8)$$

subject to  $0 \leq n_\sigma^i \leq N_\sigma^i$  and  $\sum_{\sigma \in \Sigma} n_\sigma^i = N'$ . Unfortunately, Equation (8) is difficult to maximize because the constraint that the sum of  $n_\sigma^i$  at each position must be equal causes a dependency between all of the  $n_\sigma^i$ . Since the maximum with this constraint must be smaller than the maximum without the constraint, we can maximize the following

$$\sum_{i=1}^l \left( \sum_{\sigma \in \Sigma} n_\sigma^i \log \left( \frac{n_\sigma^i}{b_\sigma \sum_{\sigma' \in \Sigma} n_{\sigma'}^i} \right) - \frac{t}{l} \sum_{\sigma \in \Sigma} n_\sigma^i \right) \quad (9)$$

The advantage of maximizing Equation (9) is that we can consider each position independently.

LEMMA 4. *The maximum of Equation (9) occurs on the boundaries of the constraints, i.e. either  $n_j^i = 0$  or  $n_\sigma^i = N_\sigma^i$ .*

PROOF. We show this for each  $n_\sigma^i$  independently and without loss of generality for  $\sigma = A$ . Let  $f(n_A^i)$  be a function we wish to maximize in terms of  $n_A^i$  where  $n_\sigma^i$ ,  $\sigma \neq A$  are fixed. Let  $\Sigma = \{A, C, G, T\}$ .

$$\begin{aligned} f(n_A^i) &= \sum_{\sigma \in \Sigma} n_\sigma^i \log \left( \frac{n_\sigma^i}{b_j \sum_{\sigma' \in \Sigma} n_{\sigma'}^i} \right) - \frac{t}{l} \sum_{\sigma \in \Sigma} n_\sigma^i \\ &= \sum_{\sigma \in \Sigma} n_\sigma^i \log(n_\sigma^i) - \sum_{\sigma' \in \Sigma} n_{\sigma'}^i \log \left( \sum_{\sigma' \in \Sigma} n_{\sigma'}^i \right) \\ &\quad - \sum_{\sigma \in \Sigma} n_\sigma^i \log(b_\sigma) - \frac{t}{l} \sum_{\sigma \in \Sigma} n_\sigma^i \end{aligned}$$

$$\begin{aligned} f'(n_A^i) &= \frac{1}{\log 2} + \log(n_A^i) - \frac{1}{\log 2} - \log \left( \sum_{\sigma' \in \Sigma} n_{\sigma'}^i \right) \\ &\quad - \log(b_A) - \frac{t}{l} \end{aligned}$$

$$f''(n_A^i) = \left( \frac{1}{n_A^i} - \frac{1}{\sum_{\sigma' \in \Sigma} n_{\sigma'}^i} \right) \frac{1}{\log 2}$$

Since  $f''(n_A^i) \geq 0$  for  $0 \leq n_A^i \leq N_A^i$ , the function is convex and the maximum must be achieved on one of the end points.  $\square$

By direct application of these two Lemmas we obtain the following Theorem.

THEOREM 1. *Given a collection of instances  $C$  with maximum nucleotide counts  $N_A^i$ ,  $N_C^i$ ,  $N_G^i$  and  $N_T^i$ . Given background probabilities  $b_A$ ,  $b_C$ ,  $b_G$  and  $b_T$ , the maximum of Equation (6) depends only on  $n_\sigma^i$  and at each position  $i$ , the maximum occurs at one of 16 points corresponding to choices of whether  $n_\sigma^i = 0$  or  $n_\sigma^i = N_\sigma^i$ . The maximum score at position  $i$  for collection  $C$ ,  $M^i(C)$ , is*

$$M^i(C) = \sum_{\sigma \in \Sigma} n_\sigma^i \log \left( \frac{n_\sigma^i}{b_\sigma \sum_{\sigma' \in \Sigma} n_{\sigma'}^i} \right) - \frac{t}{l} \sum_{\sigma \in \Sigma} n_\sigma^i$$

We use  $M(C) = \sum_{i=1}^l M^i(C)$  to denote the maximum score of the collection. We note that this theorem can be extended in a straightforward manner to incorporate constraints on  $s_\sigma^i$  such as those that come from the subspace of profiles which gives a tighter bound. We omit this extension for simplicity of presentation.

We now show how to modify Theorem 1 to take into account  $k$  sets of instances, each with a different threshold constant  $t_j$ . In each set of instances, the total number of symbols at a position in a set are  $N_{\sigma'}^{i,j}$  and the total number of instances from each set at the maximum is  $N'^j$ . We wish to maximize

$$\sum_{j=1}^k \left( \sum_{i=1}^l \sum_{\sigma \in \Sigma} n_\sigma^{i,j} \log \left( \frac{\sum_j n_\sigma^{i,j}}{b_\sigma \sum_{\sigma' \in \Sigma} \sum_{j'=1}^k n_{\sigma'}^{i,j'}} \right) - t_j N'^j \right) \quad (10)$$

subject to  $0 \leq n_\sigma^{i,j} \leq N_{\sigma}^{i,j}$  and  $\sum_{\sigma \in \Sigma} n_\sigma^{i,j} = N'^j$ .

Using the same technique as above, we can bound the maximum of Equation (10) by relaxing the constraint the all of the  $n_\sigma^{i,j}$  must sum to the same value at each position and maximize

$$\sum_{i=1}^l \sum_{j=1}^k \left( \sum_{\sigma \in \Sigma} n_\sigma^{i,j} \log \left( \frac{\sum_j n_\sigma^{i,j}}{b_\sigma \sum_{\sigma' \in \Sigma} \sum_{j'=1}^k n_{\sigma'}^{i,j'}} \right) - \frac{t_j}{l} N'^{i,j} \right) \quad (11)$$

subject to  $0 \leq n_\sigma^{i,j} \leq N_{\sigma}^{i,j}$  and  $\sum_{\sigma \in \Sigma} n_\sigma^{i,j} = N'^{i,j}$ .

Using the same analysis as in Lemma 4, the maximum value for each position occurs on the boundary. That is, the maximum occurs when  $n_\sigma^{i,j} = N_{\sigma}^{i,j}$  or  $n_\sigma^{i,j} = 0$ . Unfortunately, while previously there were only 16 cases, in this case there are  $2^{4k}$  possibilities. To check all of them is impractical.

If we sort the substrings with respect to  $t_j$  such that  $t_j > t_{j+1}$  we can make the following observation.

LEMMA 5. *At the maximum of Equation 11, if  $n_\sigma^{i,j+1} = 0$  then  $n_\sigma^{i,j} = 0$ .*

PROOF. We will show this by contradiction. Assume that at the maximum point  $n_\sigma^{i,j+1} = 0$  and  $n_\sigma^{i,j} = N_{\sigma}^{i,j}$ . Consider the second point just as the previous except that  $n_\sigma^{i,j+1} = 1$  and  $n_\sigma^{i,j} = N_{\sigma}^{i,j} - 1$ . This point is also within the constraints. For both points, the first term in Equation 11 will be equal. The second term will be greater by  $\frac{t_j - t_{j+1}}{l}$ . Since  $t_j > t_{j+1}$  the second point will be higher than the maximum which gives a contradiction.  $\square$

Using this Lemma, we can describe the  $n_\sigma^{i,j}$  for a fixed  $i$  and  $\sigma$  as being in one of  $k+1$  cases. Either they are all 0 or the  $k$  greatest of them are equal to  $N_{\sigma}^{i,j}$  and the remaining are 0. For each position this gives a total of  $(k+1)^4$  possibilities which is feasible to exhaustively check. We can approximate this value by merging close  $t_j$  together and using only the smallest  $t_j$ .

By direct application of these results above we obtain the following Theorem.

THEOREM 2. *Given  $k$  collections of instances  $C = \cup C_k$  with maximum nucleotide counts  $N_A^{i,j}$ ,  $N_C^{i,j}$ ,  $N_G^{i,j}$  and  $N_T^{i,j}$  and thresholds  $t_j$  such that  $t_j > t_{j+1}$ . Given background probabilities  $b_A$ ,  $b_C$ ,  $b_G$  and  $b_T$ , the maximum of Equation (11) depends only on  $n_\sigma^{i,j}$  and at each position  $i$ , the maximum occurs at one of  $(k+1)^4$  points corresponding to choices of whether  $n_\sigma^{i,j} = 0$  or  $n_\sigma^{i,j} = N_{\sigma}^{i,j}$  and if  $n_\sigma^{i,j+1} = 0$  then*

$n_{\sigma}^{i,j} = 0$ . The maximum score at position  $i$  for collection  $C$ ,  $M^i(C)$ , is

$$M^i(C) = \sum_{j=1}^k \sum_{\sigma \in \Sigma} n_{\sigma}^{i,j} \log \left( \frac{\sum_j n_{\sigma}^{i,j}}{b_{\sigma} \sum_{\sigma' \in \Sigma} n_{\sigma'}^{i,j}} \right) - \frac{t_j}{l} \sum_{\sigma \in \Sigma} n_{\sigma}^{i,j}$$

## 6. MITRA-PSSM

Efficient algorithms for discovering consensus patterns take advantage of the fact that the majority of the space of patterns does not contain patterns of interest. Algorithms such as SPELLER[16] and MITRA[8] are branch and bound algorithms that attempt to rule out large portions of the space of patterns by ruling out prefixes of patterns that can not lead to patterns of interest.

This idea can be applied to our search for profiles. Instead of ruling out one profile subspace at a time, we attempt to rule out many profile subspaces at the same time. Given a prefix of a discrete information content pattern, we attempt to rule out all of the profile subspaces that correspond to patterns that contain that prefix. We use the notation  $AAA???$  to represent the prefix pattern which corresponds to the set of patterns with prefix  $AAA$ . Members of this set include  $AAAAAA$ ,  $AAAAAC$ ,  $AAACAA$ , etc. We define the set of instances of a prefix pattern as the union of the instances of each pattern contained in the pattern. We can compute the upper bound described in Section 5 on the scores of the union of the subspaces of profiles corresponding to the pattern prefix. If the upper bound of the score of profiles pattern  $AAA???$  is below the minimum signal threshold, we can rule out all of the corresponding subspaces.

Pattern with mismatches algorithms like SPELLER[16] and MITRA[8] efficiently keep track of how many mismatches were observed between each substring in the data and the prefix of the pattern space. The key idea behind MITRA-PSSM is that we apply the analogous idea to compute the discrete information content score between the pattern prefix and each instance in the data. We then incorporate these scores when we compute the upper bound on the profile scores using the same upper bound described in Section 5.

### 6.1 Review of SPELLER Algorithm

In this section we provide a short overview of the major ideas behind the SPELLER algorithm and focus on what is common between this algorithm and MITRA-PSSM in some cases changing the terminology to be consistent with MITRA-PSSM. The SPELLER algorithm is presented in detail in Sagot, 1998[16] and is currently the best exhaustive algorithm for finding pattern with mismatches. SPELLER efficiently solves the  $(l, m) - k$  pattern search where the goal is to recover all patterns of length  $l$ , with  $m$  mismatches that occurs  $k$  times in the data.

SPELLER performs the search for patterns with mismatches by operating on two data structures: a data trie that contains the data and a pattern trie that represents the patterns. SPELLER initially creates a data trie for the data. At each node in the data trie, SPELLER stores the number of substrings in the subtree rooted at the node.

The  $(l, m)$ -pattern search is performed by building the pattern trie in a depth first fashion. The path from the root to a node in the pattern trie represents the prefix of the pattern space. Essentially, SPELLER is “spelling out”

the patterns on this trie. At each node in the pattern trie, SPELLER keeps pointers to the valid nodes in the data trie. That is, at each node in the pattern trie at depth  $d$ , SPELLER keeps a pointers the nodes of the data trie at depth  $d$  where the prefix of the pattern matches the path from the root of the data trie with less than  $m$  mismatches. As the pattern prefix space gets explored and the corresponding depth first search of the pattern trie proceeds, SPELLER efficiently updates the set of pointers to the valid nodes by only considering the set of pointers of the parent node in the trie. At every node in the pattern trie, SPELLER sums the number of instances of the nodes pointed to in the data trie. This gives an upper bound to the number of instances of any pattern in the pattern prefix space. If this number is below  $k$ , SPELLER can rule out the pattern prefix space which corresponds to backtracking in the depth first search. On the other hand, if SPELLER reaches a depth of  $l$  and there are more than  $k$  instances at this depth, then SPELLER outputs the pattern.

### 6.2 The MITRA-PSSM Algorithm

MITRA-PSSM is similar to the SPELLER algorithm, however, MITRA-PSSM is optimized for ruling out profile subspaces corresponding to patterns.

MITRA-PSSM also preprocesses the data by constructing a data trie. At each node of depth  $d$  in the data trie, MITRA-PSSM computes  $l - d$  vectors of counts corresponding to the occurrences of each symbol at each level of the subtree rooted at the node. These counts will be used by a variant of Theorem 1 to obtain an upper bound on profile scores.

MITRA-PSSM builds a pattern trie out of the discrete information content alphabet and keeps pointers to the “relevant” nodes in the data trie. The score of a data trie node of depth  $d$  with respect to the pattern prefix,  $s$ , is defined as the discrete information content computed from the pattern prefix and the path from the root to the node. Let  $w_{max}$  be the maximum weight in any position in the discrete information content alphabet. The maximum possible score for any pattern in the prefix space and any substring rooted at the trie node is  $s + (l - d)w_{max}$ . If the maximum score is less than  $t - l\Delta t$  then this substring is not valid for the pattern prefix. This is the analogous condition to having  $m + 1$  mismatches with pattern prefixes in SPELLER.

Since each pointer has a different corresponding path to the root, it has a different value for the discrete information content of its prefix. If we have  $k$  pointers, we now have  $k$  sets of instances (each corresponding to a node in the data trie) each with a discrete information content  $k_j$  depending on the prefix of the pattern and the path to the root of the data trie. Each node also stores the value for the substring with the minimum threshold  $q_j$ . If we set  $t_j = k_j + q_j$ , we can apply Theorem 2 to compute an upper bound on the information content of any profile in the subspace and can eliminate the entire subspace if it is below our threshold.

### 6.3 Local Search in Profile Subspace

For the candidate subspaces, we perform a local search within the profile subspace over the instances of the pattern. We apply the EM algorithm[5] which is the same algorithm used in MEME for this local search. As described above, the EM algorithm gives no guarantees on discovering the highest scoring profile. However, our search is only performed in the

profile subspace and only over the instances of the pattern which significantly reduces the search space and the chance of converging to a local minimum. In addition, since the search is restricted to the profile subspace, the problem of a strong signal masking other signals is avoided since the search will only converge to profiles in the subspace.

Various tricks have been applied to improve the performance of the EM algorithm for motif finding. Since each of our searches is in a small subspace and in the course of motif finding we must apply the local search many times, we use a very simple and efficient version of EM.

The algorithm iterates between two steps until it converges. The first step computes a profile from the set of instances (initially this is the complete set of instances of the corresponding pattern). The profile  $S$  is computed using  $s_{\sigma}^i = \frac{n_j^i}{\sum_{\sigma' \in \Sigma} (n_{\sigma'}^i + \alpha)} + \alpha$  which is the optimal profile by Lemma 3 with the addition of a smoothing factor  $\alpha$ . The smoothing factor helps the search to avoid local optima. If  $s_{\sigma}^i$  is outside of the profile subspace, we set  $s_{\sigma}^i$  to the boundary and adjust the remaining probabilities accordingly. The second step computes the set of instances from the profile. We compute the weight matrix  $W_S$  from profile  $S$  as described above. For each instance  $x_j \in I(P)$ , if  $W_S(x_j) > 0$  then the instance is included in our set of signal instances and omitted otherwise. After iterating through the algorithm several times, the set of instances will stop changing which cause the algorithm to converge to a final profile and a set of instances. If the score of the final profile is above our threshold we report the signal.

This version of the algorithm returns at most 1 profile for each subspace in the partition. However, we can easily modify this algorithm by partitioning (and re-partitioning) the subspaces that contain valid profiles to obtain more profiles and further refine our solutions.

## 6.4 Multiple Sequences

For simplicity in presentation, in Section 2 we considered that our sample consisted of a long sequence. In the more common formulation of the motif finding problem, the sample consists of many sequences and the motif occurs at most once in each each of the sequences. Since the majority of the framework is unchanged, we point out only how our approach needs to be modified to discover motifs in this problem formulation.

In Section 2, the main difference is that now we restrict the set of hidden variables that describe whether a substring  $x_j$  was generated by the motif or by the background sample  $g_j$  such that only one of the variables can be non-zero in a sequence. If two substrings have positive scores for the profile, we only set the hidden variable with the maximum value to 1 and the remaining to 0. Similarly, when we compute the information content, only the maximum positive scoring substring per sequence contributes to the score. When we perform the local search, we take this into account and only include the highest scoring substring from each sequences in the second step of the local search algorithm.

The only other major difference in our approach is that in Section 5, we compute the upper bound in a different manner. When we consider the 16 possible points for each position where the maximum can occur, we can only use 1 substring from each sequence. This reduces the number of instances and reduces the upper bound.

## 7. BIOLOGICAL EXPERIMENTS

To validate our approach, we analyze promoters from *E. Coli* prepared identically to the samples in Eskin *et. al*, 2003[7]. The sample contains close to 250,000 nucleotides and we know that there are many different strong signals in this sample. The goal of this validation is to demonstrate that MITRA-PSSM is efficient enough to use in practice and can process a large sample.

We consider patterns of length 8. Since we know the patterns contain two conserved regions separated by unconserved spacing we perform a preprocessing step to the substrings extracted by a sliding window described in [8] to remove the spacers. The background probabilities for the sample are: ( $A = 0.26, C = 0.22, G = 0.23, T = 0.27$ ) and we apply a third order Markov background model to the sample. We use the discrete information content alphabet from the Figure 2 using the symbols

$\Sigma_{DIC} = \{A, C, G, T, a, c, g, t, M, R, W, S, Y, K, N\}$  giving us a total of 15 symbols for our patterns. A pattern corresponds to a subspace of profiles as follows. We use a threshold of 9.229 which corresponds to  $P(S) = \frac{1}{1000}$ . In our partition, we use  $\Delta t = .5$  which is enough to cover the space of profiles.

The sample was processed in a about 65 minutes using MITRA-PSSM on a 750MHz computer. Although this is slower than a traditional profile algorithm, we have some guarantees that the signal is the strongest signal. The algorithm discovered many strong signals. The top scoring signal is shown in Table 2 which corresponds to the most significant biological signal in *E. Coli*, the CRP signal.

Organism	Discrete IC Pattern	Number of Instances	Discrete IC Score
<i>E. Coli</i>	TGAt-4-aTCA	465.588	206

**Table 2: Top scoring profile of length 8 based after iterating over possible separation distances for *E. Coli*. The discrete information content pattern is shown for the signals (due to space limitations) although MITRA-PSSM recovered a profile.**

Unlike previous algorithms for generating profiles, these signals are guaranteed to be close to the actual highest scoring signals. Furthermore, a single run of MITRA-PSSM recovered *all* of the signals as opposed to other algorithms that must mask each strong signal and reapply the algorithm to obtain the next signals.

## 8. CONCLUSION

We have presented a unified framework for motif finding which represents patterns as discrete approximations of profiles. The main advantage of our framework is that it motivates the MITRA-PSSM algorithm which uses a consensus patterns approach to discover profiles, taking advantage of the efficiency and guarantees of the discrete consensus pattern algorithms while preserving the expressiveness of the representation of profiles. MITRA-PSSM has the additional advantage that it can find multiple strong profiles and the strongest profiles do not mask the remaining strong signals.

However, the algorithms presented in this paper are merely the first steps in taking advantage of the framework. The framework opens the door for research in many directions. We highlight several of the promising directions below.

The current framework uses an underlying maximum like-

likelihood model. One direction of future work is to extend the framework to handle *maximum a posteriori* (MAP) likelihood and Bayesian priors.

A promising direction is to design approximation algorithms in the style of [4] within the discrete information content framework. The analysis presented here gives a new angle for analyzing an approximation algorithm for this problem.

Another direction is to devise alternative methods for partitioning the space of profiles. If we can reduce the number of subspaces, we can significantly improve our algorithms. Similarly, we may want to introduce a hierarchical partitioning scheme which potentially allows us to rule out larger subspaces yet focus more narrowly where signals are likely to occur. Another direction is to explore partitioning based on other criteria than patterns with mismatches such as random projections [4].

Finally, we can take into account more constraints when computing the upper bounds on the scores of profiles. This will give tighter upper bounds on the scores and allow more aggressive pruning increasing the efficiency of the algorithm.

MITRA-PSSM is available for public use via webserver at <http://www.calit2.net/compbio/mitra/>.

## 9. ACKNOWLEDGMENTS

The author is very thankful to the following people for useful comments: N. Friedman, M.-F. Sagot, B. Chor, P. Pevzner, and the anonymous RECOMB referees.

## 10. REFERENCES

- [1] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51, 1995.
- [2] Yoseph Barash, Gill Bejerano, and Nir Friedman. A simple hyper-geometric approach for discovering putative transcription factor binding sites. In *Proceedings of WABI*, 2001.
- [3] O.G. Berg and P. H. von Hippel. Selection of DNA binding sites by regulatory proteins. *Trends Biochem Sci.*, 13:207–211, Jun 1998.
- [4] J. Buhler and M. Tompa. Finding motifs using random projections. In *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology (RECOMB01)*, pages 69–76, 2001.
- [5] A. P. Dempster, N. Laird, and D. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, ser. B*, 39:1–38, 1977.
- [6] Alain Denise, Mireille Rgnier, and Mathias Vandenbergert. Assessing the statistical significance of overrepresented oligonucleotides. In *Proceedings of WABI*, 2001.
- [7] E. Eskin, U. Keich, M.S. Gelfand, and P.A. Pevzner. Genome wide analysis of bacterial promoter regions. In *Proceedings of the 2003 Pacific Symposium on Biocomputing*, 2003.
- [8] E. Eskin and P. A. Pevzner. Finding composite regulatory patterns in dna sequences. In *Special Issue Proceedings of the Tenth International Conference on Intelligent Systems for Molecular Biology (ISMB-2002) Bioinformatics.*, pages 1:S354–63, 2002.
- [9] D. J. Galas, M. Eggert, and M. S. Waterman. Rigorous pattern–recognition methods for DNA sequences. *Journal of Molecular Biology*, 186:117–128, 1985.
- [10] G. Hertz and G. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15:563–577, 1999.
- [11] JD Hughes, PW Estep, S Tavazoie, and GM Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *saccharomyces cerevisiae*. *Journal of Molecular Biology*, 296(5):1205–14, 2000.
- [12] U. Keich and P. A. Pevzner. Finding motifs in the twilight zone. In *Proceedings of the 6th International Conference on Computational Molecular Biology (RECOMB 2002)*, 2002.
- [13] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
- [14] A. Neuwald, J. Liu, and C. Lawrence. Gibbs motif sampling: Detection of bacterial outer membrane repeats. *Protein Science*, 4:1618–1632, 1995.
- [15] P. A. Pevzner and S. Sze. 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, 2000.
- [16] M. Sagot. Spelling approximate or repeated motifs using a suffix tree. *Lecture Notes in Computer Science*, 1380:111–127, 1998.
- [17] E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From promoter sequence to expression: A probabilistic framework. In *Proceedings of the 6th International Conference on Research in Computational Molecular Biology (RECOMB)*, 2002.
- [18] S. Sinha and M. Tompa. Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research*, 30:5549–5560, 2002.
- [19] S. Sze, M. S. Gelfand, and P. A. Pevzner. Finding subtle motifs in DNA sequences. In *Proceedings of Pacific Symposium on Biocomputing*, pages 235–246, 2002.
- [20] M.S. Waterman, R. Arratia, and D.J. Galas. Pattern recognition in several sequences: consensus and alignment. *Bulletin of Mathematical Biology*, 46:515–527, 1984.