A Flexible Motif Search Technique Based on Generalized Profiles

Philipp Bucher
Kevin Karplus
Nicolas Moeri
Kay Hofmann

January 24, 1996

Philipp Bucher
Swiss Institute for Experimental Cancer Research
CH-1066 Epalinges
Switzerland
pbucher@isrec-sun1.unil.ch
41(21)692-5892

Kevin Karplus
Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
USA
karplus@cse.ucsc.edu
1(408)459-4250

Nicolas Moeri
Département de mathématiques
École Polytechnique Fédérale
CH-1015 Lausanne
Switzerland
moeri@dma.epfl.ch
41(21)693-2595

Kay Hofmann
Swiss Institute for Experimental Cancer Research
CH-1066 Epalinges
Switzerland
khofmann@isrec-sun1.unil.ch
41(21)692-5892

ABSTRACT

A flexible motif search technique is presented which has two major components:

1. a generalized profile syntax serving as a motif definition language
2. a motif search method specifically adapted to the problem of finding multiple instances of a motif in the same sequence.

The new profile structure, which is the core of the generalized profile syntax, combines the functions of a variety of motif descriptors implemented in other methods, including regular expression-like patterns, weight matrices, previously used profiles, and certain types of hidden Markov models (HMMs).

The relationship between generalized profiles and other biomolecular motif descriptors is analyzed in detail, with special attention to HMMs. Generalized profiles are shown to be equivalent to a particular class of HMMs, and conversion procedures in both directions are given. The conversion procedures provide an interpretation for local alignment in the framework of stochastic models, allowing for clear, simple significance tests.

A mathematical statement of the motif search problem defines the new method exactly without linking it to a specific algorithmic solution. Part of the definition includes a new definition of disjointness of alignments.

Keywords: Profiles, hidden Markov models, HMMs, PROSITE, motif search, local alignment, disjointness of alignments

1 Introduction

The ultimate goal of the various genome sequencing projects is to understand the information contained in a genetic program. Elucidation of the complete base sequence of an organism's genome, or of the complete amino acid sequence of its protein inventory, constitutes only the first step towards this goal. The real challenge lies in the interpretation of these data by automatic procedures. Understanding genetic information in the scientific sense means the ability to predict the biological function of a base sequence through application of explicit rules.

The degeneracy of genetic coding mechanisms constitutes the major difficulty in this endeavor. This problem arises both at the level of gene and at the level of protein sequence. For instance, gene expression signals having the same function can exhibit a remarkable degree of sequence variation. Likewise, protein domains having similar 3D-structures may vary greatly in amino acid sequence. Despite this surprising diversity, groups of biologically related sequences usually do share some common properties. The totality of these common properties is called a *sequence motif*.

The role of a motif search technique in gene and protein function prediction is to decompose a large sequence into smaller subsequences constituting elementary structural modules or control units of elementary physiological processes. In a typical application, a new sequence of unknown function is compared against a database of many known motifs. A technique suitable for this purpose has two clearly distinguishable but operationally interdependent components. The first one is a motif descriptor or motif definition language, used to describe the motif; the second is a search method used to locate instances of the already defined motif in a particular sequence.

The motif search technique described here is the result of an attempt to conceptually unify and to combine the functions of many seemingly different approaches into a single method. Although it has been developed to support a recent format extension of the PROSITE data bank [Bairoch, 1993], it is designed as a general tool applicable in many other contexts.

The central component of the new motif definition language for PROSITE is a motif descriptor called a *generalized profile*. Accessory syntactic features control search options and other operations pertinent to motif-based sequence interpretation to make up a motif definition language called *generalized profile syntax*. A more detailed description of the generalized profile syntax together with biological examples can be found in [Bucher and Bairoch, 1994].

One objective of this paper is to define the search method for generalized profiles by an exact formulation of the mathematical problem, leaving no ambiguities to its implementation by a specific algorithm. (Efficient algorithms to solve the problem will be presented elsewhere.) A second goal of the paper is to clarify the relationships between various biomolecular motif descriptors, in particular between generalized profiles and the recently introduced hidden Markov models (HMMs), hoping that a better understanding of these relationships will facilitate communication between research communities and interoperability of research methodologies in the field.

The rest of this paper is divided into five major sections: Section 2 surveys the different motif descriptors that have been used, Section 3 describes the structure of the generalized profiles now used in PROSITE, Section 4 shows the equivalence between generalized profile alignments and Viterbi paths in a class of hidden Markov models (HMMs), Section 5 gives a description of the motif search problem, and Section 6 gives comparisons using HMMs and generalized profiles to classify Swiss-Prot into globins and non-globins.

An appendix is provided to present in detail the algorithm used to compute optimal alignment scores for general profiles. Since this algorithm is almost identical with the dynamic programming algorithms used for sequence, profile, and HMM alignment, it can be skipped by most readers.

2 Survey of biomolecular motif descriptors

A *motif descriptor* is a data structure used to define a sequence motif. Frequently used biomolecular motif descriptors include consensus sequences, weight matrices, and profiles. A motif definition based on such a descriptor may serve various functions in a motif search operation. A common capacity of all motif definitions is that they define a subset of potentially interesting sequences, either in an exact or probabilistic way. In addition, they may assign a score to a potential motif match or define a specific alignment between a sequence and an intrinsic model.

An *exact word* by itself does not qualify as a motif, but provides a didactically useful starting point to develop the hierarchical classification system of motif descriptors shown in Figure 1. One way of introducing sequence variation into an exact word is by allowing a set of alternative residues to occur at certain positions. The resulting motif descriptor is referred to as *consensus sequence with degenerate positions*. Another way of introducing sequence variation is by allowing a small number of mismatches to occur, irrespective of position. This leads to a so-called *consensus sequence with mismatches*. The two strategies can be combined into a more general type of consensus sequence.

Consensus sequences with degenerate positions and consensus sequences with mismatches represent different classes of motif descriptors. The former is a qualitative descriptor which identifies members of a sequence set. The latter is a quantitative descriptor which assigns a distance measure (the number of mismatches relative to the consensus) to each sequence of the corresponding length class. Only in conjunction with a cut-off value does such a consensus sequence define a subset of the sequence space. However, a cut-off value is typically considered a parameter of a search method rather than a parameter intrinsic to the motif definition.

A *weight matrix* is a more flexible type of quantitative motif descriptor containing weights or scores for each residue at each position. The total score assigned to a sequence of the same length is the sum of corresponding residue scores over all positions. A weight matrix score usually reflects similarity rather than distance. Weight matrices have been applied with great success to a variety of gene control signals mediated by sequence-specific DNA binding proteins (for example, see [Staden, 1984], and for reviews see [Stormo, 1988, Claverie, 1994]). The power of weight matrices results from their capacity to distinguish between mismatches of varying degrees of severity.

The *regular-expression* used in the PROSITE data bank [Bairoch, 1993] can be viewed as an extension of a consensus sequence with degenerate positions. Each position of such a pattern can be occupied by any residue from of specified set of acceptable residues, and in addition can be repeated a variable number of times within a specified range. Moreover, the pattern syntax provides features to anchor a pattern at the beginning or at the end of a sequence. PROSITE patterns are qualitative motif descriptors, like consensus sequences with degenerate positions, but differ from the former in an important way. They are variable-length motif descriptors assigning membership to a sequence class through the intermediate of an alignment between the sequence and the motif. Because different alignments are possible, a single sequence can match a pattern in different ways, as illustrated by the example in Figure 2. This raises the question whether the notion of a motif instance should be applied to a sequence matching the motif in one or several ways, or to a specific alignment between a sequence segment and the motif. The latter solution seems more appropriate because individual positions of PROSITE patterns are often associated with specific biological functions mapped to the sequence via the alignment. In such cases, two alternative alignments represent two biological hypotheses which can be tested by experiment.

The *flexible patterns* described by [Barton and Sternberg, 1990] combine elements of weight matrices and PROSITE patterns. This type of pattern consists of an alternating series of residue positions and gaps. Each residue position contains a set of weights for each residue of the sequence alphabet. The gaps define length ranges for spacer segments consisting of any sequence. Flexible patterns have been presented as a method to detect weak structural similarities of protein domains. The *sequence targets* used in [Mulligan et al., 1984] to characterize and locate *E. coli* promoters are very similar to flexible patterns. The only extension is a scoring scheme for variable-length spacer segments.

Flexible patterns and sequence targets are the simplest examples of a quantitative variable-length motif descriptor. They clearly represent generalizations of a weight matrix just as a PROSITE pattern represents a generalization of a consensus sequence with degenerate positions. The relationship between PROSITE patterns and flexible patterns is less obvious. The former contains some syntactic features that cannot be translated into the latter, for instance those allowing fixing a motif at the beginning or at the end of a sequence. Also,

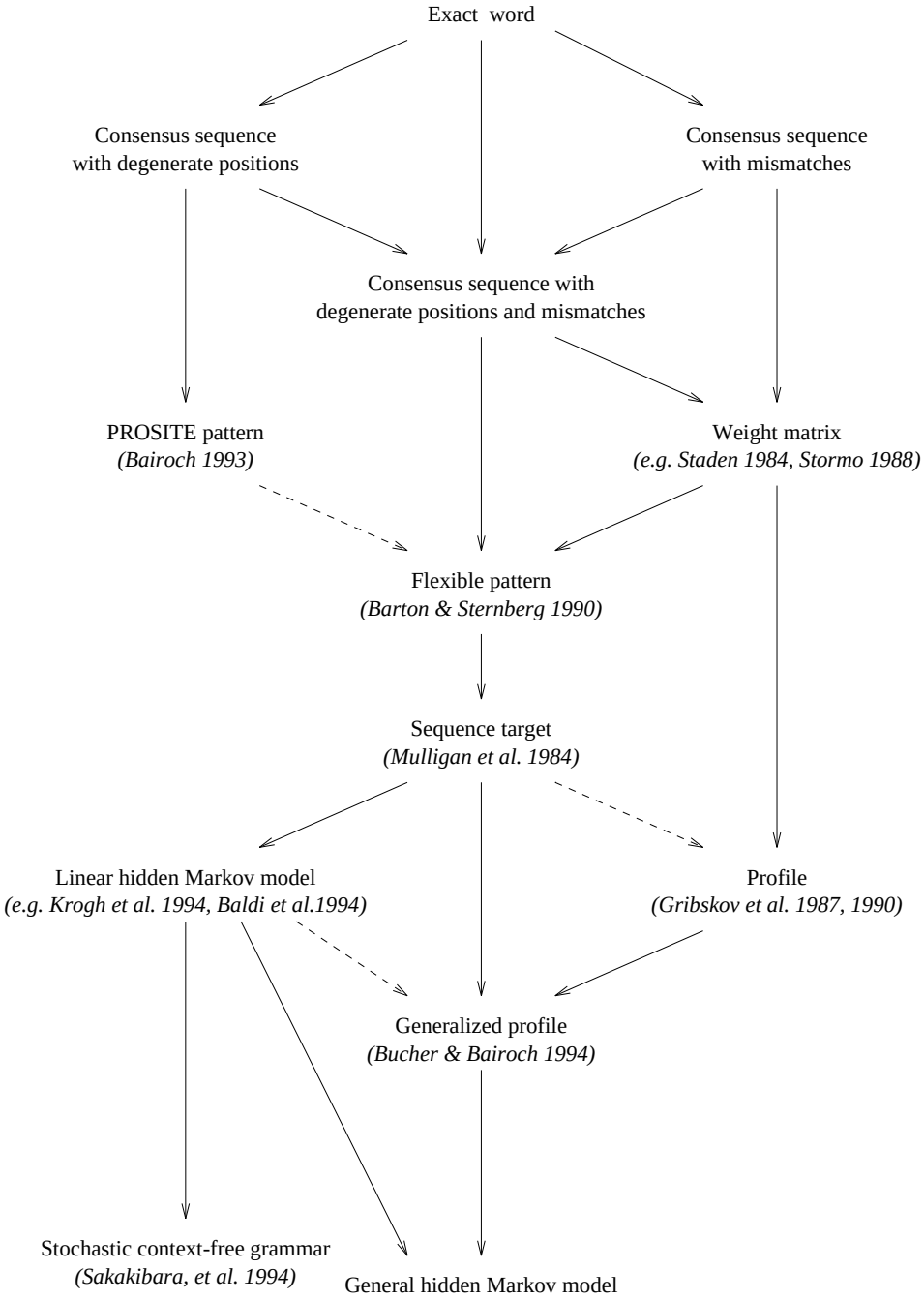

Figure 1: Relationships between various motif descriptors. Motif descriptors are arranged by increasing complexity along the vertical axis. An arrow indicates that the upper descriptor can be understood as a special case of the lower descriptor. Broken lines mean that the mapping is only approximate, or that there are exceptions which cannot be mapped to the more general descriptor (see text).

```
C -x(4,6)-[FYH]-x(5,10)- C -x(0,2)- C -x(2,3)- C -x(7,11)- C -x(4,6)-[DNEQSP]-x(2)- C
|          |            |         |         |           |           |            |
C--RKNQ----Y-RHYWSENLFQ-C ---FN--- C ---SL--- C-LNGTVHLS-C --QEKQ--- N-------TV--C


C -x(4,6)-[FYH]-x(5,10)- C -x(0,2)- C -x(2,3)- C -x(7,11)- C -x(4,6)-[DNEQSP]-x(2)- C
|          |            |         |         |           |           |            |
C--RKNQYR---H -YWSENLFQ-C ---FN--- C ---SL--- C-LNGTVHLS-C --QEKQ--- N-------TV--C
```

Figure 2: Two alternative alignments between a PROSITE pattern and a sequence. The PROSITE pattern (Acc. PS00652) describes a cysteine-rich motif of the TNFR/NGFR family. The sequence corresponds to positions 127-166 of the TNF receptor 1 precursor (Swiss-Prot Acc. P19438). The figure illustrates the fact that the same sequence segment can match a variable-length motif in various ways, representing alternative biological hypotheses.

PROSITE patterns permit variable number repetitions of any kind of positions whereas flexible patterns and sequence targets restrict this possibility to spacer positions. In practice, most PROSITE patterns are convertible into flexible patterns, because the incompatible features are rarely used.

The *profiles* introduced by [Gribskov et al., 1987, Gribskov et al., 1990] implement the idea of aligning a fixed-length weight matrix to variable-length sequences allowing for gaps in either component. The structure of a profile is very similar to that of a weight matrix. Each position contains, in addition to a complete set of residue weights, two numbers defining a linear gap penalty function for insertions and deletions starting at this position. Profiles are typically searched for with a local alignment algorithm similar to the one introduced by [Smith and Waterman, 1981]. The parameters of a profile are usually derived from a multiple sequence alignment [Gribskov et al., 1990], with or without inclusion of secondary structure information [Lüthy et al., 1991], but can also be derived from a 3D-structure model [Bowie et al., 1991]. Although simpler in structure, profiles constitute a higher level of generality than flexible patterns or sequence targets.

Recently, hidden Markov models (HMMs) of a specific architecture (here called *linear hidden Markov models*) were introduced to molecular biology [Haussler et al., 1993, Baldi et al., 1994]. These models resemble previously described motif descriptors in that they also assign a number, in this case a probability, to a specific alignment of a sequence to the model. The architectures proposed contain a higher number of parameters per length than profiles, allowing for a more flexible treatment of deletions and insertions. In this respect, they are more general than profiles. From another perspective, these architectures are more restrictive because they do not implement local alignment scoring modes. There are, however, simple modifications to hidden Markov models that allow a very close equivalence with generalized profiles, as will be shown in Section 4.

In fact, a motif description based on any of the more restrictive descriptors can be represented by a generalized profile, but the conversion procedure is not always as simple as the conversion from HMMs presented here.[1]

The generalization of the linear hidden Markov models to generalized profiles does not exhaust the possibilities of HMMs, as general (non-linear) HMMs are also useful motif descriptors [Karplus, 1994, Fujiwara et al., 1994]. Furthermore, *stochastic context-free grammars* (SCFGs) generalize linear HMMs in a different way, and have been found useful for characterizing RNA motifs [Sakakibara et al., 1994]. This paper will concentrate on generalized profiles and the equivalent HMMs, not exploring more general HMMs and SCFGs.

In summary, biomolecular motif descriptors fall into four subclasses: *qualitative, fixed-length*; *quantitative, fixed-length*; *qualitative, variable-length*; and *quantitative, variable-length*. The most general case is a quantitative, variable-length motif descriptor assigning a similarity score to all possible alignments of all sequences to the motif. Qualitative motif descriptors fit into this concept by assuming that they assign the same maximal value to all motif instances. Fixed-length motifs can be integrated by pretending that they assign very low scores (written as $-\infty$) to alignments of sequences not belonging to the corresponding length class.

[1] The detailed techniques for converting the restrictive descriptors to generalized profiles are available from Philipp Bucher.

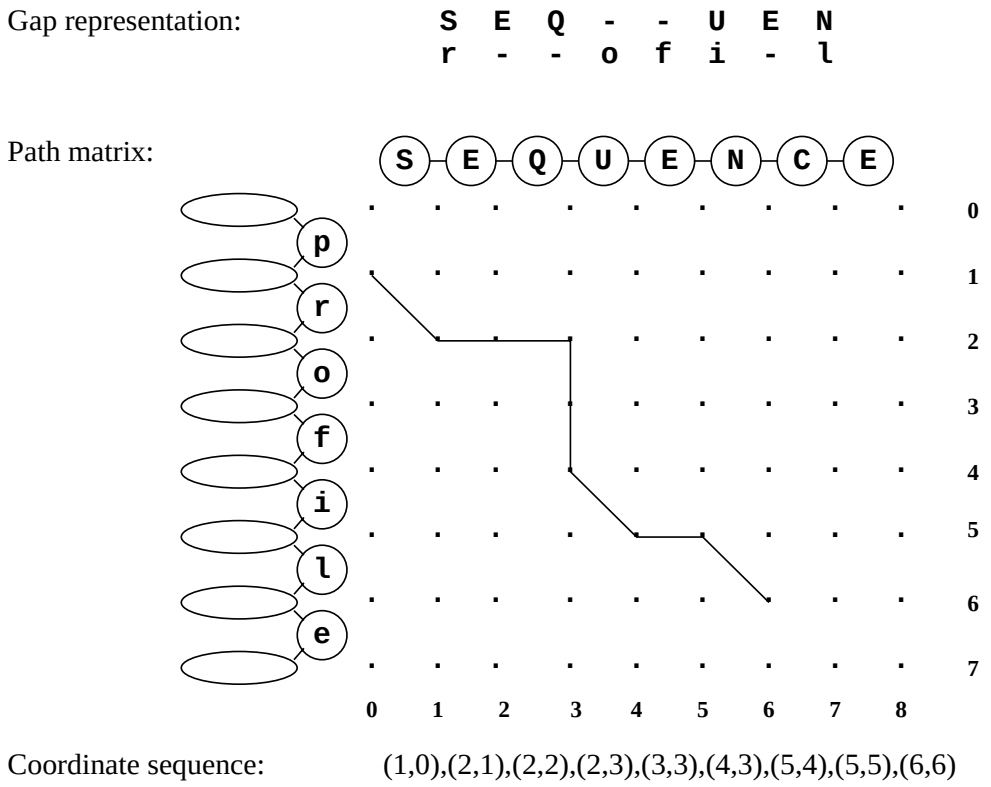

Figure 3: Three representations of a profile-sequence alignment. Note that each column in the gap representation corresponds to an elementary segment of the alignment path shown in the middle. The coordinate sequence at the bottom is a numeric representation of the alignment.

3 The structure of a generalized profile

The goal of generalized profiles is to combine the functions of all simpler motif descriptors surveyed in Section 2. It follows from this objective, and from the specific concept of a motif instance introduced before, that the function of a generalized profile will be that of an alignment scoring device. The notion of a profile-sequence alignment is thus central to its design and the properties of such an alignment largely determine its structure.

3.1 Profile-sequence alignments

The definition of a profile-sequence alignment is not as obvious as it might appear. Different alignment concepts have been introduced to molecular sequence analysis and still coexist in this field. The specific alignment type upon which generalized profiles are based must therefore be specified. Its characteristic features are highlighted in Figure 3 by means of three alternative representations of an alignment example. The gap representation shows the alignment as it would appear in a computer program output or in a scientific publication. The path matrix diagram represents the alignment as a path through a 2D coordinate system. The coordinate sequence is a numeric representation of this path. This representation is the basis of the alignment definition given in Section 3.3.

The most obvious disparity between different alignment types is that between *local alignment* and *global alignment*. A local alignment can begin and end anywhere in the profile and anywhere in the sequence, while a global alignment must begin and end at the edges of the sequence and the profile. The alignment shown in Figure 4 is local, since it does not use all of the profile. It is natural that generalized profiles are based on local alignments since local alignments are more general, including global alignment as a special case. Various

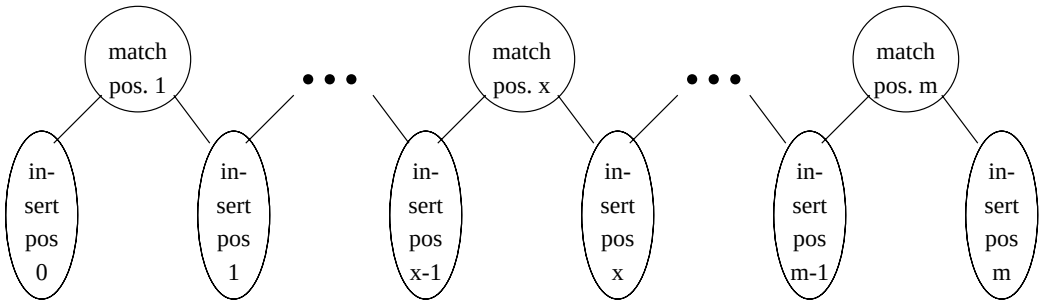

Figure 4: Structure of a generalized profile. The schematic representation defines the numbering conventions applying to profile components. The match and insert positions, represented by circles and ellipses, can be thought of as boxes containing position-specific parameters for alignment scoring.

restrictions on the parameters of a generalized profile can be placed to get global alignment or any of several other intermediate alignment styles between local and global, as described in Table 2.

Another ambiguity in the sequence alignment concept concerns the succession of different alignment components: matches, deletions, and insertions. The classical method of Needleman and Wunsch [Needleman and Wunsch, 1970] prohibits direct transitions between insertions and deletions. This restriction is also built into the profile alignment algorithm described by Gribskov *et al.* [Gribskov et al., 1990]. Most other methods, however, do allow such configurations (for example, [Sankoff, 1972, Sellers, 1974, Smith and Waterman, 1981]). The alignment example in Figure 3 makes clear that generalized profiles also allow such configurations.

Finally, there is an ambiguity concerning the representation of an alignment by a coordinate sequence. The approach chosen by us is to list every coordinate of an alignment path. A more commonly used, but less precise representation lists only the coordinates pertaining to match steps (diagonal segments of the alignment path). Using the match-only approach, the alignment shown in Figure 3 could be defined by only three coordinate pairs. The effects of the two alternative representations on the combinatorial complexity of the alignment space have been analyzed by Waterman [Waterman, 1989]. The assumption underlying the less precise approach is that the order of adjacent gaps representing pairs of unmatched sequence segments is not important. This is justified in a pairwise sequence alignment where insertions and deletions are weighted symmetrically by an external scoring system. However, in the case of a profile-sequence alignment where all weights are provided by the profile in a position-specific way, the location of insertions could affect the alignment score. Thus, the use of the more precise representation is indicated.

3.2 Parameters of a generalized profile

The structure of a generalized profile is schematized in Figure 4. It consist of an alternating sequence of match and insert positions, starting and ending with an insert position. The match positions are analogous to the letters of a sequence. The insert positions have no obvious counterpart in the sequence. The path matrix diagram in Figure 3 makes clear why two types of positions are needed. The horizontal segments of the alignment path, representing insertions relative to the profile, fall between consecutive match positions and thus must be scored by numbers accommodated between match positions.

The most basic parameters of a profile are the length and the sequence alphabet. The alphabet determines the exact number of parameters per insert and match position. Together with the length, it also determines the alignment space for which a similarity score is defined.

Let's define an alignment formally using the coordinate-pair notation:

Definition 1: *An* alignment *between a profile of length m and a sequence of n letters $a_1, \ldots, a_n$ is given by an ordered set of coordinate pairs:*

$$\{(x_0, y_0), (x_1, y_1), \ldots, (x_{l-1}, y_{l-1}), (x_l, y_l)\}$$

Scores for match position x				
$m_x(a)$				Match score for amino acid a
d_x				Delete extension score
Scores for insert position x				
$i_x(a)$				Insert score for amino acid a
$\hat{b}_x$				External initiation score
$\tilde{b}_x$				Internal initiation score
$\hat{e}_x$				External termination score
$\tilde{e}_x$				Internal termination score
$t_{b\to d,x}$	$t_{b\to m,x}$	$t_{b\to i,x}$	$t_{b\to e,x}$	Transition scores
$t_{d\to d,x}$	$t_{d\to m,x}$	$t_{d\to i,x}$	$t_{d\to e,x}$	
$t_{m\to d,x}$	$t_{m\to m,x}$	$t_{m\to i,x}$	$t_{m\to e,x}$	
$t_{i\to d,x}$	$t_{i\to m,x}$	$t_{i\to i,x}$	$t_{i\to e,x}$	

Table 1: The generalized profile has many score parameters, summarized here. Match position x is between insert positions $x-1$ and x, as shown in Figure 4. Which transition cost is used in each insert position is determined by the alignment path, as explained in the text. If the position subscript x is omitted in specifying a parameter, then the parameter is assumed to be identical for all positions.

Each x coordinate represents an insert position in the profile ($0 \le x_k \le m$) and each y coordinate represents a position between consecutive residues in the sequence ($0 \le y_k \le n$). Furthermore, adjacent coordinate pairs must have one of the following relationships:

match $x_{k+1} = x_k + 1$ *and* $y_{k+1} = y_k + 1$

insert $x_{k+1} = x_k$ *and* $y_{k+1} = y_k + 1$

delete $x_{k+1} = x_k + 1$ *and* $y_{k+1} = y_k$.

An extension step corresponds to an alignment path segment joining two consecutive coordinates. Three types of extension steps are distinguished: *match steps* associate one profile match position with one residue of the sequence and correspond to diagonal segments of the path; *insert steps* associate one profile insert position with one residue of the sequence and correspond to horizontal segments of the alignment path; *deletion steps* represent profile match positions not associated with a sequence residue and correspond to vertical segments of the alignment path. There is one insert extension score and one match extension score per residue at each insert and match position, respectively. In addition, there is a residue-independent deletion extension score at each match position.

A state transition occurs between any two consecutive extension steps, as well as at the beginning and at the end of the alignment. The possible states of an alignment are *begin, match, insert, deletion*, and *end*, symbolized by letters *b, m, i, d,* and *e*, respectively.

There are 16 different types of state transition scores for all combinations of $\{b, m, i, d\} \times \{m, i, d, e\}$. For reasons of completeness, we included a $b \to e$ transition score defining a position-specific score for empty alignments. The $b \to d$, and $d \to e$ scores are only useful in conjunction with a global alignment mode as defined in Table 2.

The scores applying to the beginning and to the end of the alignment are called *initiation* and *termination scores*. Each insert position contains two types of initiation scores, an internal and an external one. The first one applies to alignments starting at the beginning of the sequence, the second to alignments starting at a sequence internal position. The site-specificity of the external and internal termination score is analogous. The primary function of the initiation and termination scores is to encode different alignment modes (see Table 2).

The complete list of parameters contained in a generalized profile is given in Table 1. The exact function of each parameter is defined by the mathematical definition of the alignment score given in Section 3.3.

There is some redundancy in the parameters allowing for alternative representations of mathematically equivalent profiles. For example, the initiation scores and the state transition from b are distinct, but any change made to one of the scores $\tilde{b}_x$, $\hat{b}_x$, $t_{b\to d,x}$, $t_{b\to m,x}$, $t_{b\to i,x}$, or $t_{b\to e,x}$ can be compensated by changes in the other scores to get exactly the same total score for every alignment. This redundancy will be exploited in

alignment starts		alignment ends			constrained scores							
profile	seq.	profile	seq.	mode name	$k=0$		$1 \le k \le m-1$				$k=m$	
					$\hat{b}_0$	$\tilde{b}_0$	$\hat{b}_k$	$\tilde{b}_k$	$\tilde{e}_k$	$\hat{e}_k$	$\tilde{e}_0$	$\hat{e}_0$
any	any	any	any	local								
left	left	any	any	left-anchored local	$-\infty$		$-\infty$	$-\infty$				
see caption				semiglobal				$-\infty$	$-\infty$			
left	any	right	any	domain-global			$-\infty$	$-\infty$	$-\infty$	$-\infty$		
left	any	right	right	right-anchored global			$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	
left	left	right	right	global	$-\infty$		$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	

Table 2: Constraints on initiation and termination score settings to get various alignment modes. Local mode is the native mode for generalized profiles, with no parameters forced to $-\infty$. By setting certain initiation or termination scores to $-\infty$, various types of alignments can be prohibited. The first through fourth columns give the legal starting and ending positions for an alignment (starting either at the left end or anywhere in the profile or sequence and ending either at the right end or anywhere in the profile or sequence). Note that $\hat{b}_0$ and $\hat{e}_m$ are not constrained in any of the alignment modes. This table summarizes and names some of the more useful settings. A particularly useful setting is the semi-global alignment, which allows the alignment to start either at the left end of the sequence or the left end of the profile, and stop either at the right end of the profile or the right end of the sequence. This allows finding complete motifs within longer sequences and fragmentary motifs that are cut off at the beginning or end of a sequence, without getting a lot of less interesting partial matches buried in the middle of sequences.

Section 4.4 to convert generalized profiles to equivalent ones that are more easily converted to hidden Markov models.

3.3 Definition of scores for generalized profile-sequence alignments

We can compute the score for an alignment between a generalized profile and a sequence by adding up several parts: an initiation score, an extension score for each adjacent pair of coordinates, a state-transition score for each coordinate pair, and a termination score.

The initiation score is either the external or the internal initiation score (depending on whether or not we start at the beginning of the sequence) for the first position in the profile that is actually used:

$$\text{begin} = \begin{cases} \hat{b}_{x_0} & \text{if } y_0 = 0 \\ \tilde{b}_{x_0} & \text{otherwise} \end{cases} . \tag{1}$$

The extension score depends on the type of extension (match, insert, or delete), and the amino acid involved in matches or insertion:

$$\text{ext}(k) = \begin{cases} m_{x_k}(a_{y_k}) & \text{if } x_{k-1} = x_k - 1 \text{ and } y_{k-1} = y_k - 1 \\ i_{x_k}(a_{y_k}) & \text{if } x_{k-1} = x_k \text{ and } y_{k-1} = y_k - 1 \\ d_{x_k} & \text{if } x_{k-1} = x_k - 1 \text{ and } y_{k-1} = y_k \end{cases} . \tag{2}$$

The state-transition scores are determined by the type of extension on either side of the position:

$$\text{trans}(k) = t_{v_k \to v_{k+1}, x_k} \text{ for } 0 \le k \le l \tag{3}$$

where

$$v_k = \begin{cases} b & \text{if } k = 0 \\ m & \text{if } x_{k-1} = x_k - 1 \text{ and } y_{k-1} = y_k - 1 \\ i & \text{if } x_{k-1} = x_k \text{ and } y_{k-1} = y_k - 1 \\ d & \text{if } x_{k-1} = x_k - 1 \text{ and } y_{k-1} = y_k \\ e & \text{if } k = l + 1 \end{cases} . \tag{4}$$

Finally, we include either the external or the internal termination score for the last position of the profile that is actually used:

$$\text{end} = \begin{cases} \hat{e}_{x_l} & \text{if } y_l = n \\ \tilde{e}_{x_l} & \text{otherwise} \end{cases} . \tag{5}$$

The alignment score itself is defined as the sum of the above-defined components:

$$S(A) = \text{begin} + \sum_{k=1}^{l} \text{ext}(k) + \sum_{k=0}^{l} \text{trans}(k) + \text{end} .$$

With this definition of the alignment score, calculation of the optimal alignment score is a straight-forward dynamic programming algorithm, almost identical to the algorithms used for sequence alignment or HMM alignment. Refer to Appendix A for a more detailed presentation of this dynamic programming.

4 Generalized profiles are equivalent to a class of hidden Markov models

This section shows the equivalence between generalized profiles and a class of hidden Markov models. Section 4.1 explains what a hidden Markov model is, and what subset of them we are interested in, Section 4.2 looks at the relationship between probabilities of paths in a hidden Markov model and scores in a generalized profile, and Sections 4.3 and 4.4 establish the equivalence.

4.1 What is a hidden Markov model?

Hidden Markov models (HMMs) are one way of encoding information about a set of finite-length sequences over some alphabet—in our case, sequences of amino acids. They model the sequences as being generated by a stationary stochastic process—that is, they assign a probability to every possible sequence. For a good introduction to HMM techniques, see [Rabiner, 1989].

A hidden Markov model is a directed graph consisting of vertices (called *states*) and edges (sometimes called *transitions*, though we will use that term for a particular group of edges in an HMM). The HMMs we are interested in have two types of states: *letter states*, each of which has an associated probability distribution of letters from the same alphabet, and *null states*, which have no associated letters. In the diagrams for this paper, letter states are shown as square boxes and null states as circles. Also, each edge has an associated probability, with the probabilities of the edges out of any state summing to one.

An HMM has two distinguished states: the *start state* and the *stop state*, both of which are null states. The start state has no in-edges and the stop state has no out-edges.

We compute the probability of a sequence w by looking at all paths from the start state to the stop state and computing the probability of each path and the probability of the sequence given that path. The probability of a path in an HMM is just the product of the probabilities of the edges along the path. The probability of a sequence given a path is the product of the probabilities of the letters in the corresponding letter states (or zero, if the number of letter states on the path is not the same as the length of the sequence). If we call the ith letter of the sequence w_i and the ith letter state on the path l_i, then

$$P_{\text{model}}(w, \text{path } p) = \left(\prod_{\text{edges } e \in p} P(e) \right) \left(\prod_i P(w_i | l_i) \right)$$

and

$$P_{\text{model}}(w) = \sum_{\text{paths } p} P_{\text{model}}(w, p) \ ,$$

where the sum is to be interpreted as including only those paths from the start node to the stop node that have the right number of letter states.

With this definition of the probability of a sequence given an HMM, there is no hope for finding an equivalence with generalized profiles, since the profiles pick only the highest scoring path, not all possible paths. However, if we redefine our model so that the "probability" of a sequence is the maximum over all paths of the right length from start to stop, rather than the sum over such paths:

$$P_{\text{model}}(w) = \max_{\text{paths } p} P_{\text{model}}(w, p) \ ,$$

then we can find an equivalence.

The use of the maximum probability path (often called the *Viterbi path*) is quite common with HMMs, as it is cheaper to compute and provides alignment information that is not easily obtainable with the sum-of-probabilities definition. The probability assigned to a sequence by the Viterbi path in an HMM is a lower bound on the true probability (sum over all paths) assigned by the HMM.

We are interested only in a small subclass of HMMs here—those that are equivalent to generalized profiles. We'll show this class of HMMs by diagram, but first let's introduce some notation to simplify the diagrams. First, a *node* is a pair of states: a letter state called the *match state* and a null state called the *delete state*. The node will be drawn as a vertical ellipse, as shown in Figure 5. Second, a *transition* is a collection of three states (begin, insert, and end) and sixteen edges connecting two nodes, as shown in Figure 6.

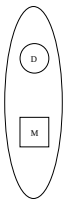

Figure 5: A node contains one match state (M) and one delete state (D).

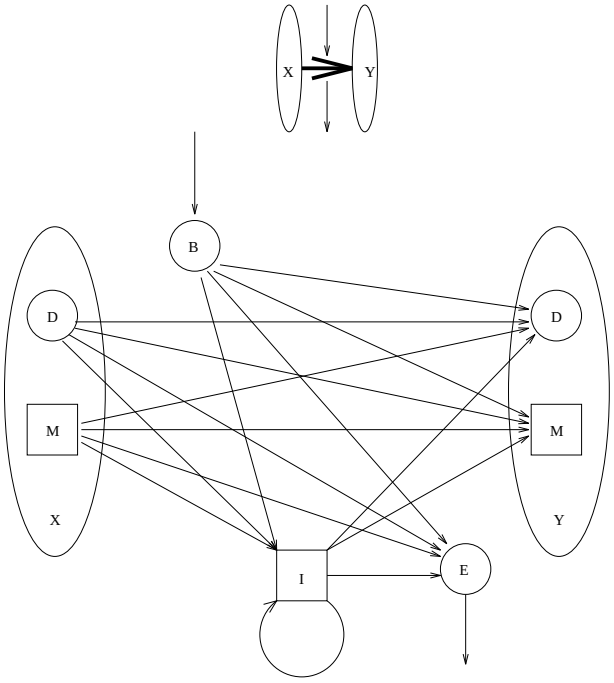

Figure 6: The top symbol (a heavy arrow connecting two nodes) stands for the transition shown below. Note that the insert state and the begin and end null states and the sixteen edges that connect them to the adjacent nodes are all part of the transition. In the text, we will refer to a particular transition, such as this one from X to Y, with an arrow $X \rightarrow Y$, and the three states in the transition as $B_{X\rightarrow Y}$, $I_{X\rightarrow Y}$, and $E_{X\rightarrow Y}$.

4.2 The null model

Before we can construct the hidden Markov Model equivalent to a generalized profile, we need to examine a little more closely what the score from the profile means.

As Altschul pointed out [Altschul, 1991], any alignment score for a sequence w can be interpreted as making an assertion about the ratio of two probabilities. If we think of our sequences as being generated by some stochastic process or model m, the score s is the logarithm of the ratio of the probability of the sequence being generated by the model $P_m(w)$ and the probability of the sequence being generated by a null model $P_\emptyset(w)$ (with some arbitrary logarithmic base z):

$$\text{score}(w) = \log_z \frac{P_m(w)}{P_\emptyset(w)} \ .$$

Note that high scores can result from sequences that aren't modeled well by the null model, as well as from sequences that are well modeled by model m.

There is a simple significance test that can be applied to hidden Markov models (or any other modeling scheme that assigns probabilities to all sequences). Milosavlević's algorithmic significance test asserts that the probability of getting a score larger than T for sequences distributed according to the null model is less than

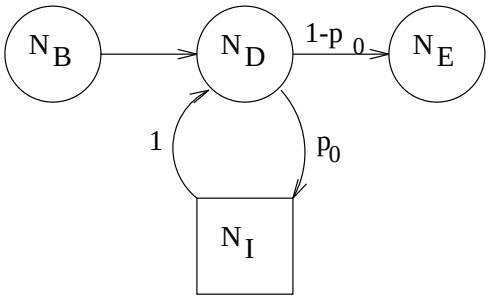

Figure 7: This is the implicit null model for a generalized profile. It generates sequences from a background distribution specified by the insert state N_I with a geometric length distribution specified by the probability p_0.

z^{-T} [Milosavljević and Jurka, 1993]. Of course, this test relies on using a reasonable null model, which is not always available.

Before we can construct an equivalent hidden Markov Model for a generalized profile, we need to know what null model was used for the generalized profile. If the null model is provided with the profile, we can use it, otherwise we can choose a standard null model. The model in Figure 7 is a reasonable null model for most profiles. The insert state N_I can be set up to match the background distribution, and looping with probability p_0 gives a geometric length distribution for the sequences. This null model (or a similar one) is implicit in many of the scoring systems used for alignment.

We can construct an HMM equivalent to a generalized profile for almost any null model of the form proposed in Figure 7, as long as N_I does not assign a zero probability to any letter, and p_0 is not too close to one. The exact constraint on p_0 will be shown when we get to the constraining step in the construction.

A future extension of the generalized profile syntax used in PROSITE will provide a way to specify null models and the logarithmic base to be used for converting probabilities to scores.

4.3 Converting an HMM to a generalized profile

The HMMs we are interested in are those that have the structure shown in Figure 8. The structure of the HMMs corresponds in an obvious way to that of a generalized profile—each node corresponds to a match position of the generalized profile, each transition to an insert position, the edges from B to the external initiation scores, the edges from B' to the internal initiation scores, the edges to E to the external termination scores, and the edges to E' to the internal termination scores. The HMM is a linear HMM, except for the copies of the null models added at the beginning and end to handle the internal initiation and termination scores.

In one direction, the equivalence between HMMs and generalized profiles is easy—we can take a linear HMM m of the appropriate form and convert it to a generalized profile G, such that for any sequence w the score generated by the G is the log probability ratio:

$$\text{score}_G(w) = \log_z \frac{P_m(w)}{P_\emptyset(w)} \ .$$

The parameters of the generalized profile are given in Table 1—all we have to do is to show how to set these parameters.

First, we set the match extension scores for position x to the probabilities given by the match state in node x:

$$m_x(a) = \log_z \frac{P(a|M_x)}{P_{N_I}(a)p_0} \ ,$$

where $P_{N_I}(a)$ is the background distribution given by the null insert state N_I. We set the insert scores similarly from the insert states:

$$i_x(a) = \log_z \frac{P(a|I_{x\to x+1})}{P_{N_I}(a)p_0} \ ,$$

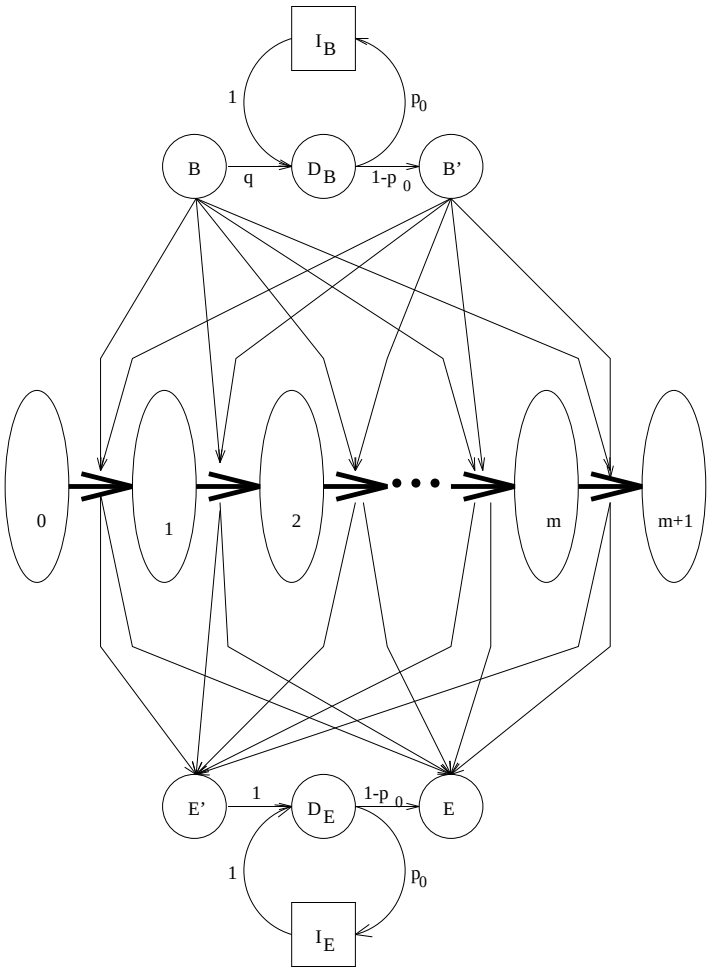

Figure 8: A linear HMM that is equivalent to a generalized profile. The nodes on the left and right ends are dummies (the states in them are never part of a path from the start state B to the stop state E). They are present only to create the adjacent transitions (especially the insert states). Note the copies of the null model to join the true start state B to the internal start B' and the internal stop E' to the true stop state E.

and we set all delete-extension scores to zero. (Remember that the notation $I_{x\rightarrow x+1}$ means the insert state in the transition from node x to node $x+1$ of the HMM.) Note that we have incorporated p_0 into the match and insert scores, rather than into some transition score. We do this for convenience in keeping track of the number of times p_0 gets included, since the edge in the null model is traversed once for each letter in the sequence.

The transition scores are set to the log of the probability of the corresponding edges in the HMM.

The external initiation scores are set to the log of the probabilities of the edges from B:

$$\hat{b}_x = \log_z P(B \rightarrow B_{x\rightarrow x+1}) \ ,$$

and the external termination scores are set similarly

$$\hat{e}_x = \log_z \frac{P(E_{x\rightarrow x+1} \rightarrow E)}{1-p_0} \ .$$

The extra $1-p_0$ term in the external termination score is to correct for the final edge of the null model from N_D to N_E.

The only slightly tricky part is handling the internal initiation and termination scores. Since the generalized profile does not have any structure corresponding to the part of the HMM between B and B', we have to make

sure that all sequences generated by paths from B to B' would have the same score (log probability ratio in the HMM and the null model), and add that score to the log probability of the edge from B' to get the profile score.

If we set the internal initiation scores to

$$\tilde{b}_x = \log_z \left(P(B' \rightarrow B_{x \rightarrow x+1}) q(1-p_0)\right) \ ,$$

and the internal termination scores to

$$\tilde{e}_x = \log_z \left(P(E_{x \rightarrow x+1} \rightarrow E')\right)$$

then the scores for all paths will have the proper correspondence to the probabilities in the HMM.

There is one minor difference between the generalized profiles and the HMMs: generalized profiles do not allow the internal initiation scores to be used at the beginning of the sequence, nor the internal termination scores to be used at the end of the sequence, but the HMMs do have paths from B through B' (and through E' to E) that match zero-length strings, allowing the use of internal initiation (and termination) scores at the ends of the sequence.

The HMM-to-generalized-profile conversion works as long as

$$P(B' \rightarrow B_{x \rightarrow x+1}) q(1-p_0) < P(B \rightarrow B_{x \rightarrow x+1})$$

and

$$P(E_{x \rightarrow x+1} \rightarrow E')(1-p_0) < P(E_{x \rightarrow x+1} \rightarrow E) \ ,$$

since then the paths $B \rightarrow D_B \rightarrow B' \rightarrow x$ and $x \rightarrow E' \rightarrow D_E \rightarrow E$ will always have lower probability than the direct paths $B \rightarrow x$ and $x \rightarrow E$ and so never appear on the Viterbi path. These constraints on the probabilities are equivalent to the following constraints on the scores in generalized profiles: $\tilde{b}_x < \hat{b}_x$ and $\tilde{e}_x < \hat{e}_x$.

Those constraints, which are met by all existing generalized profiles, insure that the internal scores would never be used at the ends of the sequences even without the prohibition in the definition, and so the prohibition can be removed from the definition without changing the meaning of the existing profiles. If the constraints are not met, the generalized profile can have some rather non-intuitive behavior, preferring incomplete matches to complete ones, to avoid the low external scores.

Note that profiles constructed from linear HMMs, such as those constructed by SAM [Hughey and Krogh, 1995], will have $-\infty$ for all initiation scores except in position 0 and for all termination scores except in position m, because the corresponding edges do not exist (equivalently, have probability zero) in SAM's HMMs. The free-insertion modules of SAM correspond roughly to the beginning and ending null models of our HMMs, and so all four of the parameters $\tilde{b}_0$, $\hat{b}_0$, $\tilde{e}_m$, and $\hat{e}_m$ can be used. From Table 2, we can see that the profiles constructed from SAM HMMs use domain-global alignment mode (or global mode, if free-insertion modules are not used).

4.4 Generalized profile to HMM

If a generalized profile has been created by conversion from an HMM, we can easily reverse the construction to re-create the HMM. We first look up or select a null model, then construct the graph for the HMM as in Figure 8, and finally assign probabilities in the obvious way, reversing the equations of Section 4.3.

We can do the conversion no matter what non-zero background probabilities we assume for the null model, but there are some constraints on the loop probability p_0, imposed by the insert scores in the generalized profile.

We would like to do this construction of an HMM for any generalized profile, but if we just apply the formulas blindly we can end up with "probabilities" on the edges of the HMM that don't add up to one at some states. If all the sets of "probabilities" computed by reversing the formulas of Section 4.3 add up properly to one, then we say that the generalized profile is in *normal form*.

We can also end up with probabilities of zero for various edges or for character probabilities in match or insert states, where the profile has scores of $-\infty$. In order to ensure that all parts of the Markov model are reachable, one generally requires non-zero probabilities on all edges. The simplest solution is to replace the $-\infty$ scores with a large negative number, so that no probability is zero, but the prohibited edges or matches have such a low probability that they will never appear on any Viterbi path.

Since the insert and delete costs in profiles are often rather arbitrarily scaled, we may want to choose different bases for the logarithms in the transition scores than for the logarithms in the match and insert scores. Such a translation would not preserve the scores of paths, but might be useful for translating profiles whose gap costs have not been properly chosen.

If the generalized profile G is not in normal form, we have to find an equivalent generalized profile G' that is. The equivalence cannot preserve scores exactly—the best we can do is to guarantee that there is some constant c such that

$$\text{score}_G(w) = c + \text{score}_{G'}(w) \ .$$

The important observation to make is that the scores we are interested in are scores of paths through the generalized profiles—we can change the parameters of the profile arbitrarily, as long as the scores of all paths remain the same.

There are some obvious ways that we can modify scores—for example, we can subtract a constant from all match scores in position x and add that constant to all the transition scores $t_{*\to m,x-1}$ or to all the transition scores $t_{m\to *,x}$.[2] Since every alignment that does a match at position x must pass through a transition into M at x and a transition out of M at x, these changes in the parameters make no difference to any path score. The same operation can be done for insert scores and delete extension scores.

We can also apply this operation twice to transfer a constant from all transitions $t_{m\to *,x}$ to $t_{*\to m,x-1}$. For insertions the transition score $t_{i\to i,x}$ is unchanged by the corresponding transfer, since the score is associated with both an in-edge and an out-edge of the insert state.

We can use this operation of pushing constants backwards through the profile to convert the generalized profile into normal form.

First, we want to normalize the match scores so that the corresponding letter states in the HMM will have probabilities that sum to one. Since $P_{M_x}(a) = z^{m_x(a)} P_{N_I}(a) p_0$, we can accomplish this by subtracting $\log_z \left(p_0 \sum_b z^{m_x(b)} P_{N_I}(b)\right)$ from the match scores in position x, and adding it each of the transition scores $t_{*\to m,x-1}$. The insert scores are similarly normalized, and the delete scores are eliminated by adding them to the $t_{*\to d}$ scores.

Next, starting at the end of the model, we normalize all transition scores so that the probabilities of the out-edges of each state in the HMM sum to one, moving the normalizing constant from the out-edges to the in-edges.

There is only one tricky part to this normalization: handling the insert loops correctly. Since moving a constant through an insert loop doesn't change the score of the loop edge itself, we have the constraint that the transition score $t_{i\to i,x}$ after the insert letter scores have been normalized must already be the log of some probability p. We normalize the remaining transition scores $t_{i\to m,x}$, $t_{i\to d,x}$, and $t_{i\to e,x}$ so that the corresponding probabilities sum to $1-p$.

In order for the loop edge to have an acceptable probability after normalizing the letter scores, we have the constraint (in the scores of the original profile)

$$t_{i\to i,x} + \log_z \left(p_0 \sum_b z^{i_x(b)} P_{N_I}(b) \right) < 0 \ .$$

Since the null model is not provided in the generalized profile, and we are forced to guess one, these constraints on the insert loops can be viewed as upper bounds on our guess for p_0:

$$p_0 < \frac{\sum_b z^{i_x(b)} P_{N_I}(b)}{z^{t_{i\to i,x}}} \ .$$

Note: the constant q in the HMM for the probability of doing any internal initiation is set automatically by sweeping the constants back and normalizing the probabilities of the edges out of B. The cycles for the copies of the null model (I_B, D_B and I_E, D_E) are unchanged by pushing constants back through them, just as the self-loops on the insert states were unchanged.

[2] The symbol * is used to indicate any of the legal extension types.

We are left at the end of the normalizing process with unnormalized transition probabilities out of B, and no place to push the normalizing constant back to. This is the constant c mentioned as being unavoidable in the conversion process.

The conversion process just described relies heavily on the HMM being a left-right HMM, with no cycles in the graph except the loops on the insert states and the D_B and D_E cycles. If we allow circular profiles (merging nodes 0 and $m + 1$ of Figure 8), then we have to impose other constraints on the scores to ensure that an alignment path even exists.

5 The motif search problem

So far we have spoken only about alignments and scores, and have not talked about the real problem we are trying to solve: finding biologically meaningful instances of motifs. In a typical application, one is interested in the following questions:

1. What sequences contain the motif?
2. How many times does the motif occur in the sequence?
3. Where are the motif instances located?
4. How similar are these motif instances to the motif?
5. How can the motif instances be aligned to the motif?

It is important to recognize that these questions cannot be fully answered by formulating the motif search problem as a classification problem, which answers only the first question. In fact, a major shortcoming of published database search algorithms using profiles or hidden Markov models is that they are designed to find only the single best alignment between the model and a sequence, or to compute only a single value to assess membership of a sequence family. The advantage of the motif search method defined here is that it provides a complete answer to the above questions.

There are two reasons why the search for a biomolecular sequence motif is not a trivial task, even if the motif is accurately defined by an appropriate descriptor. The first one is that genetic texts, in the form of nucleotide sequences or translated into protein, do not contain any obvious punctuation signals. As a consequence, delineation of functional subsequences and classification of these subsequences must proceed simultaneously. The second reason is that biological sequence motifs, of the same or of different types, may occur in partially overlapping fashion. The high degeneracy of many motifs favors such an arrangement. However, the physical overlap constraints vary greatly between different motifs. For instance, the same protein sequence cannot simultaneously participate in the formation of two autonomous structural domains. By contrast, a short DNA sequence can simultaneously be part of two protein recognition sites, located on opposite sides of the double helix. Therefore, a generally applicable motif search technique must deal with the overlap problem in a flexible way.

5.1 Motif search problem for generalized profiles

Given the function of generalized profiles, namely to assign a score to an alignment, it follows that the result of an elementary motif search operation involving one profile and one sequence must have the form of a set of alignments with corresponding similarity scores. As a first approximation, the goal can be described as finding all alignments with scores higher than a prescribed cut-off value. However, in the literal sense, this is not the desired result because each alignment exceeding the cut-off value is usually surrounded by a large number of similar alignments also exceeding the cut-off value. Usually one wants such a group of alignments be represented by a single, locally optimal alignment. This can be achieved by requiring that two alignments contained in the result of a motif search operation meet a specific disjointness criterion.

The motif search problem can be more precisely stated with the following definition:

Definition 2: *The* motif search problem *is to find a set E_A of p alignments $A_1, \ldots, A_p$ given a sequence, a profile, a symmetric disjointness relationship $\diamondsuit$ between two alignments, and a cut-off value c, respecting the following conditions:*

1. *The score of each alignment of the set is greater or equal to the cut-off value: $\forall i$ with $1 \leq i \leq p$, $S(A_i) \geq c$.*
2. *The alignment A_1 is a maximally scoring alignment, that is, $\forall$ alignments B, $S(A_1) \geq S(B)$.*
3. *Any two alignments in the set are disjoint: $\forall i, j$ with $1 \leq i < j \leq p$, $A_i \diamondsuit A_j$.*
4. *No alignment of the set can be replaced by a better one without violating the disjointness condition: $\forall i$ with $1 \leq i \leq p$, $\forall B$ (if $\forall j \neq i$ $B \diamondsuit A_j$, then $S(B) \leq S(A_i)$).*
5. *No alignment whose score is greater or equal to the cut-off value can be added to the set without violating the disjointness condition: E_A is maximal in the sense of inclusion.*

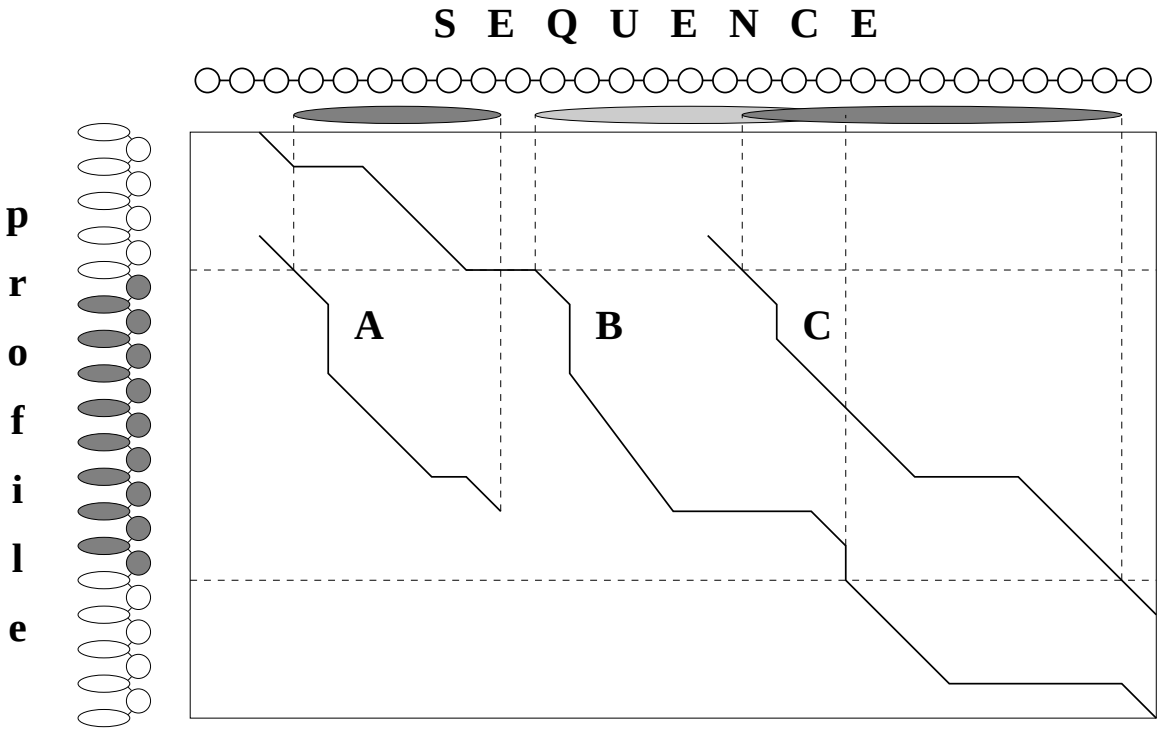

Figure 9: Geometric meaning of the disjointness definition used in PROSITE. The shaded area of the profile corresponds to the protected region. The sequence ranges mapped by three different alignments to the protected region are indicated by ellipses below the sequence. Alignments A and B are disjoint because these ranges do not overlap. By the same argument, alignments B and C are not disjoint and thus cannot appear together in the result of a motif search.

In the generalized profile syntax, the cut-off value and disjointness definition are implemented as external accessories. This is because these parameters are only loosely associated with the motif defined by the profile. In a real application, the choice of these parameters often depends on the specific purpose of a motif search application, rather than on biological properties of the motif. For instance, with a profile that does not absolutely reliably identify a motif, one may want to maximize either sensitivity or selectivity depending on the goal of the analysis. The intended effect can be achieved by an appropriate choice of the cut-off value. In certain situations, it may be desirable to see multiple suboptimal alignments of the same sequence region with the motif, even if these alignments are mutually exclusive. The number of such alignments appearing in the result can be controlled by varying the disjointness definition.

5.2 Disjointness definition used in PROSITE

The definition of the motif search problem relies on an unspecified disjointness relationship between alignments. The standard disjointness test used in PROSITE represents an adaptation of the problem of finding multiple non-intersecting best local alignments between two sequences formulated by [Waterman and Eggert, 1987].

The basic idea of disjointness is that the profile matches two separate parts of the sequence independently. The particular disjointness definition used in PROSITE declares a range of match positions, including intervening insert positions, as a *protected region* of the profile. For two alignments to be disjoint, the parts of the sequence that are aligned to positions in the protected region must not overlap. The geometric meaning of this condition is illustrated by the path matrix shown in Figure 9. Unlike the disjointness definition applied by the method for finding multiple best alignments between sequence pairs, this definition includes two adjustable parameters (the ends of the protected region) conferring remarkable flexibility to the motif search method (for a discussion of biological examples, see [Bucher and Bairoch, 1994]).

The following definitions make this concept more precise:

Definition 3: *The* protected region $[m_1 : m_2]$ of a profile consists of match positions $m_1, \ldots, m_2$ and insert positions $m_1, \ldots, m_2 - 1$.

Definition 4: *The* sequence range $[n_1 : n_2]$, mapped by alignment A to the protected region of the profile $[m_1 : m_2]$ consists of all residues mapped by A to a match or insert position of the protected region of the profile.

Definition 5: *Two alternative alignments A and A' between a sequence and a profile are* disjoint if and only if the sequence ranges $[n_1 : n_2]$, $[n'_1 : n'_2]$, mapped by A, A', respectively, to the protected region of the profile $[m_1 : m_2]$, do not have any residue in common.

If no protected region is specified, it is assumed that the entire profile is protected, and alignments are disjoint if they have no characters of the sequence in common. If an alignment results in no characters of the sequence mapping to the protected region, then the alignment is disjoint from all other alignments. Such a situation usually indicates a poorly designed profile, and it is reasonable for a computer program to reject such alignments (with a warning, of course).

5.3 Choosing sets of alignments

The calculation of the optimal alignment is straightforward (see Appendix A for details), but the algorithm for choosing sets of disjoint alignments to match Definition 2 is not obvious. This section outlines the algorithm used in PROSITE to generate reasonable sets.

The methods for finding alignment sets for motif search are adapted from efficient algorithms for finding multiple best local alignments between two sequences (for a review, see [Pearson and Miller, 1992]). The basic algorithm for finding multiple disjoint alignments is an iterative procedure that adds one alignment to the set on each iteration. The generic form of the algorithm is to, on each iteration,

1. add the best alignment in the path matrix to the alignment set, and
2. remove from the path matrix all edges that are parts of non-disjoint alignments.

(See Appendix A for explanation of the path matrix and calculation of optimal alignments.)

This algorithm is applicable to variety of disjointness definitions including the standard definition used in PROSITE and the *no-common-pair* definition introduced in [Waterman and Eggert, 1987].

Implementation of the basic methods is simple, requiring only two modifications to the algorithm used to compute the optimal alignment score. First, the recursive equations defining the score at each path matrix node have to account for the edges that have been removed from the path matrix. Second, some limited alignment information needs to be kept to make the necessary modifications after acceptance of an alignment. If the standard PROSITE disjointness definition is used, only the beginning and end points of the sequence region associated with the protected region need to be recorded.

The generic algorithm is relatively slow, requiring recomputation of the entire path matrix for each accepted alignment, but the sequence analysis literature provides many hints how to speed up this process. For example, for the no-common-pair disjointness definition, [Huang and Miller, 1991] contains a efficient algorithm not requiring storage of the entire path matrix. A similar time- and space-efficient solution for the PROSITE disjointness definition has been developed, and will be described in a subsequent paper.

Even with the most efficient methods, keeping track of alignment information slows down the computation of the optimal alignment algorithm by at least a factor of two. Since most protein domains covered in PROSITE occur in few sequences, it is advantageous to scan a protein database by first computing the optimal alignment score in the most efficient way (possibly using a co-processor), then applying the multiple motif search algorithm only to those few sequences which exceed some cut-off value.

There are two technical issues one has to be aware of when implementing this method:

1. There can be multiple best alignments having the same score at any stage of the iterative process. Thus, the statement of the problem in Definition 2 does not define a unique solution. Algorithms for finding multiple alignments between sequences handle this problem by applying *tie-breaking* rules that provide a ranking between equally scoring alignments. Any function that assigns a unique value to every possible alignment path is adequate for this purpose. It must be realized, however, that application of different tie-breaking rules may affect the result of a motif search operation in unexpected ways. It may change not only the geometry of the alignments, but also the corresponding similarity scores, or even the total number

of alignments included in the result. Detailed descriptions of motif-search algorithms should explicitly state the tie-breaking rules used—the paper that describes the efficient algorithm used for PROSITE will contain this information as well.

2. An alignment may have no sequence characters associated with the protected region. With Definition 5, such an alignment is disjoint from any alignment, including itself, and so the algorithm will loop infinitely, since no edge can be removed from the path matrix. a reasonable way to deal with this exception is to restart the entire computation with an higher cut-off value that excludes the faulty alignment or with a larger protected region.

name	length	description
GLB_ASCSU	338	two globin domains
GLB_PSEDC	333	two globin domains
FHP_CANNO	387	flavohemoprotein
FHP_YEAST	399	flavohemoprotein
HMPA_ALCEU	403	flavohemoprotein
HMPA_ECOLI	396	flavohemoprotein
HMPA_VIBPA	394	flavohemoprotein
HBB1_UROHA	146	fragments (contains 24 Xs)
HBA2_PLEWA	133	fragment
HBAZ_MESAU	102	fragment
HBA1_UROHA	90	fragment (contains 4 Xs)
HBB_DASVI	87	fragment
HBA2_UROHA	40	fragment
GLB2_GLYDI	45	fragment (contains 2 Xs)
GLB1_LAMSP	41	fragment
HBB2_UROHA	19	fragment
HBB_OVIMU	145	(contains 12 Xs)
GLBH_CAEEL	159	debatable globin (included)
GLB_TETPY	121	debatable globin (rejected)
GLBN_NOSCO	118	debatable globin (rejected)
GLB_PARCA	116	debatable globin (rejected)

Table 3: Special sequences in Swiss-Prot release 31 for the globin-recognition problem. The flavohemoproteins are much longer than normal for globins, but contain a globin domain at one end. The nine annotated fragments vary considerably in length. There are only four proteins of debatable status, a globin-like protein from *C. elegans* and a group of three heme-binding proteins from protozoa and cyanobacteria. The former is classified as a true globin in this study, the latter three are rejected for the reasons given in [Takagi, 1993].

6 Using generalized profiles and hidden Markov models to identify globins

In this section, we will show how the interconversion of generalized profiles and HMMs can lead to better results than using either method alone, and how the understanding of the equivalence can be used to improve existing search tools. To illustrate these points, we use generalized profiles and HMMs to separate globins from non-globins in the Swiss-Prot database.

6.1 The globin-recognition problem

The globin-recognition problem has often been used as an example to demonstrate the efficiency of methods to detect weak sequence similarities. Obvious advantages of this system are the relative abundance of sequences (676 complete globin sequences in Swiss-Prot release 31), and the availability of several high resolution 3-D structures representing diverse subfamilies. Moreover, the problem provides the necessary degree of difficulty as standard pairwise sequence comparison methods usually fail to detect significant similarities between distant members. Finally, the evaluation of different methods is greatly facilitated by the high degree of certainty in the classification of globins.

The 676 sequences include a few unusual sequences, and exclude a few sequences that others may consider globins. Table 3 lists the unusual sequences.

In previous studies of this kind (for example, [Gribskov et al., 1987, Krogh et al., 1994]), the globin-recognition problem was treated as a sequence classification problem. However, such an approach simplifies the underlying biological problem since globin folding units, like most other protein structural domains, occasionally occur as multiple copies in the same polypeptide chain. As the new search techniques described in Section 5 allow

automatic identification of multiple motif instances in the same sequence, the performance of the newly derived globin profiles will be evaluated by the number of correctly identified globin *domains* rather than globin *sequences*.

The parallel tests performed with the HMM search algorithm have to classify sequences, since the SAM package [Hughey and Krogh, 1995] used does not have the ability to report multiple alignments.

The results obtained with the two search techniques are nevertheless comparable because the protein database searched contains only two sequences with multiple globin domains.

6.2 Construction of HMMs and generalized-profiles

In order to separate the evaluation of different model construction methods from the evaluation of different search techniques, we constructed some profiles with methods designed for profiles, and some HMMs with methods designed for HMMs, then used our conversion techniques to convert each to the other. We used both generalized-profile search methods and HMM search methods to evaluate each of the models.

We constructed two profiles using existing profile-construction methods: Profile-3d and Profile-333. The same construction technique was used for both, but Profile-3d started from a structural alignment of seven globins [Bashford et al., 1987], while Profile-333 started from a multiple alignment of 333 randomly chosen globins that was produced automatically using ClustalW [Thompson et al., 1994a]. The set of sequences included one of the two-domain globins (GLB_ASCSU), two of the flavohemoproteins (FHP_YEAST and HMPA_VIBPA), and five fragments.

The multiple alignments were directly converted to generalized profiles using the current method used for constructing PROSITE profiles. This method involves gap excision [Thompson et al., 1994b] and a symmetric gap weighting mode made possible by the new parameters of generalized profiles. The sequences in the alignments were weighted with the method of [Sibbald and Argos, 1990]. The match tables were created with a $10 \log_{10}$-scaled BLOSUM-45 matrix [Henikoff and Henikoff, 1992], and position-specific gap weights were created using parameters recommended in [Lüthy et al., 1994]. The initiation and termination scores were set to zero, except for the ones that need to be $-\infty$ to get semiglobal alignment (see Table 2). The search for multiple domains set the protected region of the profile to all but the first and last five positions of the profile.

The two profiles were converted to equivalent HMMs by the method of Section 4.4. The logarithmic base z for the conversion was estimated by examining the average entropy of the match positions in a natively trained HMM (HMM-333) and setting z so that the average entropy of match positions in the converted HMM was the same as in the native one.

We tried various ways of deriving z using just statistics about the profile scores, but were unable to come up with an appealing way to set z. For example, we tried scoring a large set of random sequences created by window-shuffling Swiss-Prot release 29 with a window size of 20 [Pearson, 1990]. The logarithmic base z was chosen so that the probability of a score larger than t was approximately z^{-t} for large values of t. This did not work particularly well—indeed the value of z obtained this way was a factor of two too small.

If too large a value is chosen for z, the HMM search using all paths is essentially the same as one using Viterbi paths only, since the optimal alignment will have a much higher probability than slightly poorer ones. If too small a value is chosen for z, the HMM search will give much too high a probability to poor alignments, and not classify sequences well. The ranks of individual alignment paths are not changed by the choice of z (in particular, the Viterbi path remains the same), only the probabilities assigned to the alignments are changed.

The parameters for the null model assumed in the conversions are shown in Table 4. The probabilities of the letters correspond to their frequencies in Swiss-Prot release 31, and the self-loop probability to an expected sequence length of 333.

Two HMMs were constructed using the SAM program buildmodel and a set of training sequences. For HMM-333, buildmodel was started with a random model that it constructed, while for HMM-prof-3d buildmodel was started with an HMM converted from Profile-3d.

One error was made in building the models—the insert positions were given flat distributions (based on the models that SAM creates by default) rather than background probability distributions. This caused the HMMs to perform poorly, as compositionally biased sequences could get much higher scores in the HMM than in the null model, even without good alignments. For example, if the null model that the HMM was compared with used the background frequencies, then the histidine-rich, non-globin HRPX_PLALO is much too highly scored, since histidine has a lower probability in the null model than the HMM. If the flat distribution is used as a null model,

parameter	probability	parameter	probability	parameter	probability
P(A)	0.0760	P(C)	0.0176	P(D)	0.0529
P(E)	0.0628	P(F)	0.0401	P(G)	0.0695
P(H)	0.0224	P(I)	0.0561	P(K)	0.0584
P(L)	0.0922	P(M)	0.0236	P(N)	0.0448
P(P)	0.0500	P(Q)	0.0403	P(R)	0.0523
P(S)	0.0715	P(T)	0.0581	P(V)	0.0652
P(W)	0.0128	P(Y)	0.0321	p_0	0.9970

Table 4: Parameters of the null model (see Figure 7) used for conversion of profiles to HMMs and for normalizing scores of HMMs.

then the alanine-rich, non-globin TOLA_ECOLI is scored too high, since the alanines score better in the match positions of the profile than in the null model, even in poor alignments.

The training sequences were the 333 randomly chosen sequences used for creating Profile-333. Since SAM does not yet support sequence weighting, the training set was given a crude weighting by hand: those sequences in the training set that scored poorly with an early version of the model were duplicated, and those that scored extremely poorly were replicated four times. This training set was frozen early in the process of developing the models, and the replicated sequences may not in fact be the low-scoring ones in the final model.

The models were trained using the free-insertion modules provided by SAM, but scoring using Krogh's ad hoc length normalization scheme resulted in very poor performance for the HMMs. This surprised us, since the models worked well with profile search.

We managed to get good performance on the HMMs by removing the free insertion modules and patching the models to use the natural scoring system:

$$\text{score}(w) = \log_z \frac{P_m(w)}{P_\emptyset(w)} \ .$$

This patch consisted of removing the free insertion modules, changing all insertion letter tables to the background frequencies (simultaneously putting the null model in the first and last insert position and fixing the incorrect probability distribution for the other insert states), and changing the self-loop probabilities of the first and last insert states to 0.997.

Some special adjustment was made to initial and final transitions, to give the HMMs a fair chance of finding fragments. The transition from the start state to the first delete state was given a probability of 0.04, from the start state to the first insert 0.96×0.997 and from the start state to the first match state 0.04×0.997. Transistions from the final match state to the end state were given probability 0.008 and from the final delete state to the end state 0.04. The 0.04 probabilites for the transitions to and from delete were chosen to approximately match the frequency of fragments in the training set. Note that all this ad hoc patching could easily be incorporated into the standard conversion method.

The patch was applied to the HMMs converted from generalized profiles as well, and it is the patched HMMs whose performance is reported in Section 6.3.

It would be useful if SAM supported training to maximize the score difference between the model and the null model, rather than maximizing the score in isolation. The concept of "free-insertion modules" could be replaced with "null-model insertion modules", so that these copies of the null models are kept identical to the null model throughout training.

6.3 Evaluation of HMMs and generalized-profiles

The classification results from using HMM search techniques and generalized-profile search techniques on the same model (summarized in Table 5) are comparable. For both search methods the scores were normalized $-\log_{10} \frac{P_m(w)}{P_\emptyset(w)}$. The generalized-profile scores were normalized by fitting an extreme-value distribution to the high scores obtained from scoring a window-shuffled version of Swiss-Prot release 29 [Pearson, 1990]. The HMM

Search using generalized profile				
	Profile-3d	Profile-333	HMM-333	HMM-prof-3d
lowest globin score	7.11	5.34	8.25	9.44
highest non-globin score	6.89	7.78	7.78	7.65
gap	0.22	-2.54	0.47	1.79
mean of 5 highest non-globin scores	6.57	7.11	7.15	6.98
number of globin sequences(domains) missed	0	3(4)	0	0
number of fragments missed	2	2	2	2
Search using HMM				
	Profile-3d	Profile-333	HMM-333	HMM-prof-3d
lowest globin score	2.49	6.08	9.26	10.75
highest non-globin score	6.39	8.73	6.58	7.35
gap	-3.90	-2.65	2.68	3.40
mean of 5 highest non-globin scores	6.32	7.69	6.39	6.68
number of globin sequences missed	8	6	0	0
number of fragments missed	5	4	4	4
number of false positives	13	12	12	11
old PROSITE profile				
number of globin domains missed	0			
number of fragments missed	1			

Table 5: Results of various attempts to model globins using HMMs and generalized profiles. The scores reported are all normalized to be $-\log_{10}$ probability, and for the size database searched, scores larger than about 7 should be significant. The "gap" reported is the difference in score between the lowest complete globin domain and the highest non-globin. Because the HMM program SAM has difficulty with X characters, high scoring non-globins with many Xs were classified as false positives, and the highest scoring non-globin was chosen from among the sequences with fewer than 30 Xs. The performance of the old hand-constructed PROSITE motif description is included for comparison.

results were normalized by subtracting off the scores of the null model, and changing the base of the logarithm from e to 10.

The following differences are noticeable;

- SAM is not able to handle wild-card characters correctly, assigning them a probability of one instead of the more natural choice $2^{-\text{entropy of table}}$. As a result, sequences with many wildcards get much too high a score. There are 23 sequences in the data base with 30 or more X characters, and about half of them cause problems with the HMM classification. These sequences have not been counted in choosing the highest scoring non-globin, but are listed in the table as false positives.
- SAM does a poorer job in finding fragments than searches using generalized profiles. This difference comes from a difference in alignment modes. The SAM package [Hughey and Krogh, 1995] used for linear HMM search supports only the family and domain models introduced by Krogh [Krogh et al., 1994], which represent special cases of the more general model proposed in Section 4 of this article. The models available with SAM can only do global and domain-global searches, which makes finding the fragments difficult, while the profiles were all set to semiglobal alignment mode (see Table 2).
- The HMM scoring system generally creates a larger gap between the worst-scoring globin and the best-scoring non-globin with native HMMs, indicating a clearer separation of the classes.

Given the similarity in search results, is there any point to having both HMM and generalized-profile software? Definitely—our best model HMM-prof-3d was generated by a hybrid method: First we created a generalized profile from a structural alignment, then converted to an HMM, did HMM training on a larger set of unaligned sequences, and finally converted back to a generalized profile for searching with semiglobal alignment.

Our conversion methods lead to new insights into how to normalize HMM scores appropriately, and suggested several minor improvements to the SAM HMM tool.

7 Conclusions

We have presented a unified formalism to describe biomolecular sequence motifs and motif search algorithms. Underlying this formalism is a specific motif concept with biological and mathematical connotations. Central to the concept is that a motif instance is a specific alignment of a sequence region with a motif descriptor, not just the sequence region alone. This motif concept can be applied to groups of protein or nucleic acid sequences that share some common sequence features, because of either a common function or a common evolutionary origin.

In presenting this formalism, we made a clear distinction between three different problems related to sequence motifs.

- Deriving a sequence motif from initial data. We have not addressed this problem in detail here, relying instead on existing techniques for profiles and HMMs.
- Describing the shared sequence properties constituting a sequence motif. The generalized profiles represent a generalization of many of the previously used motif descriptors, including a certain class of HMMs. The relationship between these descriptors has been analyzed in detail and conversion procedures for HMMs and generalized profiles have been given.
- Locating and identifying instances of an already defined motif in functionally uncharacterized sequences. We defined several different alignment modes. One of them, semiglobal alignment, is perhaps more useful than traditional modes such as local or global alignment.

  We emphasized the distinction between the classification and the motif search problem. Methods addressing the first one are primarily useful for evaluating the validity of motif models on sequence regions of known function, but not in situation where the total number, as well as the the start and end points of individual motif instance are not known in advance.

The motif search problem stated here relies on four essential concepts. The first is the motif descriptor itself, alternatively called a model. The second is the definition of the alignment, which in conjunction with a motif description and a target sequence, defines the search space. The third is the scoring function for alignments. The fourth is a disjointness definition, which together with the alignment scoring function, serves to define a non-redundant set of potentially interesting motif instances. We kept the first three components fixed and left the specific definition of the disjointness open. The basic outline of this formulation can provide a framework for defining motif search methods based on more complex descriptors such as general HMMs and SCFGs, if the definitions of the alignment and disjointness relation are appropriately modified.

The benefits from having a common formalism are manifold: Being able to convert motif descriptions derived by many different techniques obviously eases the design of versatile motif databases and search software. A formal framework allows concise description of the behavior of motif search software independently of the specific algorithm used.

Besides this, the study of the relationship between different methods can lead to better understanding of the underlying theories and improvement of the existing tools. We have exemplified this aspect by comparing linear hidden Markov models with generalized profiles, at a theoretical level as well as by a case study. This comparison has been productive in many ways—the two most important achievements being the design of a more effective search technique for HMMs, and the formulation of local alignment in the theoretical framework of hidden Markov models.

Acknowledgements

Philipp Bucher was responsible for the definition and analysis of generalized profiles. Kevin Karplus was responsible for establishing the exact equivalence of paths in HMMs and alignments in generalized profiles. Nicolas Moeri was responsible for the formal definition of the alignment score, the motif search problem, and Appendix A. Kay Hofmann was responsible for building the generalized profile search tools and for setting parameters of the profile in converting from alignments.

The work on the relationship between hidden Markov models and generalized profiles was stimulated by recent discussions with Pierre Baldi and David Haussler.

This work was supported in part by grant 31-37687.93 from the Swiss National Research Foundation and grant MIP-9423985 from the U. S. National Science Foundation.

References

[Altschul, 1991] Altschul, S. F. (1991). Amino acid substitution matrices from an information theoretic perspective. *JMB*, 219:555–565.

[Bairoch, 1993] Bairoch, A. (1993). The PROSITE dictionary of sites and patterns in proteins, its current status. *NAR*, 21:3097–3103.

[Baldi et al., 1994] Baldi, P., Chauvin, Y., Hunkapillar, T., and McClure, M. (1994). Hidden Markov models of biological primary sequence information. *PNAS*, 91:1059–1063.

[Barton and Sternberg, 1990] Barton, G. J. and Sternberg, M. J. (1990). Flexible protein sequence patterns: A sensitive method to detect weak structural similarities. *JMB*, 212(2):389–402.

[Bashford et al., 1987] Bashford, D., Chothia, C., and Lesk, A. M. (1987). Determinants of a protein fold: Unique features of the globin amino acid sequence. *JMB*, 196:199–216.

[Bowie et al., 1991] Bowie, J. U., Lüthy, R., and Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170.

[Bucher and Bairoch, 1994] Bucher, P. and Bairoch, A. (1994). A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. In *ISMB-94*, pages 53–61. AAAI Press.

[Claverie, 1994] Claverie, J.-M. (1994). Some useful statistical properties of position-weight matrices. *Computers and Chemistry*, 18(3):287–294.

[Fujiwara et al., 1994] Fujiwara, Y., Asogawa, M., and Konagaya, A. (1994). Stochastic motif extraction using hidden Markov model. In *ISMB-94*, pages 121–129. AAAI Press.

[Gribskov et al., 1990] Gribskov, M., Lüthy, R., and Eisenberg, D. (1990). Profile analysis. *Methods in Enzymology*, 183:146–159.

[Gribskov et al., 1987] Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: Detection of distantly related proteins. *PNAS*, 84:4355–4358.

[Haussler et al., 1993] Haussler, D., Krogh, A., Mian, I. S., and Sjölander, K. (1993). Protein modeling using hidden Markov models: Analysis of globins. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 1, pages 792–802, Los Alamitos, CA. IEEE Computer Society Press.

[Henikoff and Henikoff, 1992] Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *PNAS*, 89:10915–10919.

[Huang and Miller, 1991] Huang, X. and Miller, W. (1991). A time-efficient, linear-space local similarity algorithm. *Adv. Appl. Math*, 12:337–357.

[Hughey and Krogh, 1995] Hughey, R. and Krogh, A. (1995). SAM: Sequence alignment and modeling software system. Technical Report UCSC-CRL-95-7, University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064.

[Karplus, 1994] Karplus, K. (1994). Using Markov models and hidden Markov models to find repetitive extragenic palindromic sequences in *Escherichia coli*. Technical Report UCSC-CRL-94-24, University of California, Santa Cruz.

[Krogh et al., 1994] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling. *JMB*, 235:1501–1531.

[Lüthy et al., 1991] Lüthy, R., McLachlan, A. D., and Eisenberg, D. (1991). Secondary structure-based profiles: Use of structure-conserving scoring table in searching protein sequence databases for structural similarities. *PROTEINS: Structure, Function, and Genetics*, 10:229–239.

[Lüthy et al., 1994] Lüthy, R., Xenarios, I., and Bucher, P. (1994). Improving the sensitivity of the sequence profile method. *Protein Science*, 3:139–146.

[Milosavljević and Jurka, 1993] Milosavljević, A. and Jurka, J. (1993). Discovering simple DNA sequences by the algorithmic similarity method. *CABIOS*, 9(4):407–411.

[Mulligan et al., 1984] Mulligan, M. E., Hawley, D. K., Entriken, R., and McClure, W. (1984). Escherichia coli promoter sequences predict in vitro RNA polymerase selectivity. *NAR*, 12:789–800.

[Needleman and Wunsch, 1970] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *JMB*, 48:443–453.

[Pearson, 1990] Pearson, W. R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.*, 183:63–98.

[Pearson and Miller, 1992] Pearson, W. R. and Miller, W. (1992). Dynamic programming algorithms for biological sequence comparison. *Methods Enzymol.*, 210:575–601.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286.

[Sakakibara et al., 1994] Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjölander, K., Underwood, R. C., and Haussler, D. (1994). Stochastic context-free grammars for tRNA modeling. *NAR*, 22:5112–5120.

[Sankoff, 1972] Sankoff, D. (1972). Matching sequences under deletion/insertion constraints. *PNAS*, 69:4–6.

[Sellers, 1974] Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM J. App. Math*, 26:787–793.

[Sibbald and Argos, 1990] Sibbald, P. and Argos, P. (1990). Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *JMB*, 216:813–818.

[Smith and Waterman, 1981] Smith, T. F. and Waterman, M. S. (1981). Comparison of bio-sequences. *Adv. Appl. Math*, 2:482–489.

[Staden, 1984] Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *NAR*, 12:505–519.

[Stormo, 1988] Stormo, G. D. (1988). Computer methods for analyzing sequence recognition of nucleic acids. *Annu. Rev. Biophys. Biophys. Chem.*, 17:241–263.

[Takagi, 1993] Takagi, T. (1993). Hemoglobins from single-celled organisms. *Curr. Opin. Struct. Biol.*, 3:413–418.

[Thompson et al., 1994a] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994a). CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties, and weight matrix choice. *NAR*, 22(22):4673–4680.

[Thompson et al., 1994b] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994b). Improved sensitivity of profile searches through the use of sequence weights and gap excision. *CABIOS*, 10(1):19–29.

[Waterman, 1989] Waterman, M. S. (1989). Sequence alignments. In Waterman, M. S., editor, *Mathematical Methods for DNA Sequences*, chapter 3. CRC Press.

[Waterman and Eggert, 1987] Waterman, M. S. and Eggert, M. (1987). A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *JMB*, 197:723–728.