# Profile hidden Markov models

Sean R. Eddy
Dept. of Genetics, Washington University School of Medicine
4566 Scott Ave., St. Louis MO 63110 USA
eddy@genetics.wustl.edu

keywords: profiles, hidden Markov models, protein families

## Abstract

*Summary:* I review the recent literature on profile hidden Markov model (profile HMM) methods and software. Profile HMMs turn a multiple sequence alignment into a position-specific scoring system suitable for searching databases for remotely homologous sequences. Profile HMM analyses complement standard pairwise comparison methods for large scale sequence analysis. Several software implementations and two large libraries of profile HMMs of common protein domains are available. HMM methods performed comparably to threading methods in the CASP2 structure prediction exercise.

*Contact:* eddy@genetics.wustl.edu

## Introduction

Proteins, RNAs, and other features in genomes can usually be classified into families of related sequences and structures (Henikoff *et al.*, 1997). Different residues in a functional sequence are subject to different selective pressures. Multiple alignments of a sequence family reveal this in their pattern of conservation. Some positions are more conserved than others, and some regions of a multiple alignment seem to tolerate insertions and deletions more than other regions.

Intuitively, it seems desirable to use position specific information from multiple alignments when searching databases for homologous sequences. "Profile" methods for building position-specific scoring models from multiple alignments were introduced for this purpose (Taylor, 1986; Gribskov *et al.*, 1987; Barton, 1990; Henikoff, 1996). However, profiles have been less used than pairwise methods like BLAST (Altschul *et al.*, 1990; Altschul *et al.*, 1997) and FASTA (Pearson & Lipman, 1988), with the most notable exceptions being the popular BLOCKS database (Henikoff *et al.*, 1998), and the skilled use of profiles by a small band of professional protein domain hunters (Bork & Gibson, 1996).

In part, this is because the residue scoring systems used by pairwise alignment methods are supported by a significant body of statistical theory (Altschul & Gish, 1996). The probabilistic "meaning" of position-independent pairwise alignment scoring matrices is well understood (Altschul, 1991), allowing powerful scoring matrices to be derived (Henikoff & Henikoff, 1992). The statistical significance of ungapped pairwise alignment scores can be calculated analytically, and the significance of gapped alignment scores can be calculated by simple empirical procedures (Altschul & Gish, 1996; Altschul *et al.*, 1997). In contrast, profile methods have historically used *ad hoc* scoring systems. Some mathematical theory was desirable for the meaning and derivation of the scores in a model as complex as a profile (Henikoff, 1996).

Hidden Markov models (HMMs) now provide a coherent theory for profile methods. Hidden Markov models are a class of probabilistic models that are generally applicable to time series or linear sequences. HMMs have been most widely applied to recognizing words in digitized sequences of the acoustics of human speech (Rabiner, 1989). HMMs were introduced into computational biology in the late 1980's (Churchill, 1989), and for use as profile models just a few years ago (Krogh *et al.*, 1994a).

Here, I review the recent literature on profile HMM methods and related methods for modeling sequence families. Preference is given to papers appearing in the past two years, since my last review of the field (Eddy, 1996). There seem to be three principal advances. First, motif-based HMMs have been introduced as an alternative to the original Krogh/Haussler profile HMM architecture (Grundy *et al.*, 1997; Neuwald *et al.*, 1997). Second, large libraries of profile HMMs and multiple alignments have become available, as well as compute servers to search query sequences against these resources (Sonnhammer *et al.*, 1998). Third, there has been an increasing incursion of profile HMM methods into the area of protein structure prediction by fold recognition (Levitt, 1997).

Because of space limitations, some of the background I give is terse. A satisfactory introduction to HMMs and probabilistic models is beyond the scope of this review. Tutorial introductions to HMMs are available (Rabiner, 1989), including introductions that specifically include profile HMM methods (Krogh, 1998). Two recent books describe probabilistic modeling methods for biological sequence analysis in detail (Baldi & Brunak, 1998; Durbin *et al.*, 1998).

## Hidden Markov models

There are now various kinds of profile HMMs and related models, all based on HMM theory. It is useful to understand the generality and relative simplicity of HMM theory before considering the special case of profile HMMs.

A hidden Markov model describes a probability distribution over a potentially infinite number of sequences. Because a probability distribution must sum to one, the "scores" that an HMM assigns to sequences are constrained. The probability of one sequence cannot be increased without decreasing the probability of one or more other sequences. It is this fundamental constraint of probabilistic modeling (Jaynes, 1998) that allows the parameters in an HMM to have nontrivial optima.

*[Figure 1]*

An example of a simple HMM that models sequences composed of two letters $(a, b)$ is shown in Figure 1. This toy HMM would be an appropriate model for a problem in which we thought sequences started with one residue composition (a-rich, perhaps), then switched once to a different residue composition (b-rich, perhaps). The HMM consists of two *states* connected by *state transitions.* Each state has a *symbol emission* probability distribution for generating (matching) a symbol in the alphabet. It is convenient to think of an HMM as a model that generates sequences. Starting in an initial state, we choose a new state with some transition probability (either staying in state 1 with transition probability $t_{1,1}$, or moving to state 2 with transition probability $t_{1,2}$); then we generate a residue with an emission probability specific to that state (for example, choosing an $a$ with $p_1(a)$). We repeat the transition/emission process until we reach an end state. At the end of this process, we have a hidden *state sequence* that we don't observe, and a *symbol sequence* that we do observe.

The name "hidden Markov model" comes from the fact that the state sequence is a first order Markov chain, but only the symbol sequence is directly observed. The states of the HMM are often associated with meaningful biological labels, such as "structural position 42". In our toy HMM, for instance, states 1 and 2 correspond to a biological notion of two sequence regions with differing residue composition. Inferring the alignment of the observed protein or DNA sequence to the hidden state sequence is like labeling the sequence with relevant biological information.

Once an HMM is drawn, regardless of its complexity, the same standard dynamic programming algorithms can be used for aligning and scoring sequences with the model (Durbin *et al.*, 1998). These algorithms, called Forward (for scoring) and Viterbi (for alignment), have a worst-case algorithmic complexity of $O(NM^2)$ in time and $O(NM)$ in space for a sequence of length $N$ and an HMM of $M$ states. For profile HMMs that have a constant number of state transitions per state rather than the vector of $M$ transitions per state in fully connected HMMs, both algorithms run in $O(NM)$ time and $O(NM)$ space – not coincidentally, identical to other sequence alignment dynamic programming algorithms. For a modest (constant) penalty in time, very memory-efficient $O(M)$ and $O(M^{1.5})$ versions of Viterbi and Forward can also be implemented (Hughey, 1996; Tarnas & Hughey, 1998).

Parameters can be set for an HMM in two ways. An HMM can be *trained* from initially unaligned (unlabeled) sequences. Alternatively, an HMM can be *built* from prealigned (prelabeled) sequences (i.e. where the state paths are assumed to be known). In the latter case, the parameter estimation problem is simply a matter of converting observed counts of symbol emissions and state transitions into probabilities. In building a profile HMM, an existing multiple alignment is given as input. In contrast, training a profile HMM is analogous to running a multiple alignment program before building the model, and thus is a harder problem.

Training algorithms are of interest because we may not yet know a plausible alignment for the sequences in question. The standard HMM training algorithms are Baum-Welch expectation maximization or gradient descent algorithms. Gibbs sampling, simulated annealing, and genetic algorithm training methods seem better at avoiding spurious local optima in training HMMs and HMM-like models (Eddy, 1996; Durbin *et al.*, 1998; Neuwald *et al.*, 1997). Most training algorithms seek relatively simple maximum likelihood (or maximum *a posteriori*) optimization targets. More sophisticated optimization targets are used to compensate for nonindependence of example sequences (e.g. biased representation) (Eddy, 1996; Durbin *et al.*, 1998; Sunyaev *et al.*, 1998; Karchin

& Hughey, 1998), or to maximize the ability of a model to discriminate a set of true positive example sequences from a set of true negative training examples (Mamitsuka, 1996).

However, since HMM training algorithms are local optimizers, it pays to build HMMs on prealigned data whenever possible. Especially for complicated HMMs, the parameter space may be complex, with many spurious local optima that can trap a training algorithm.

In contrast to parameter estimation, a suitable HMM architecture (the number of states, and how they are connected by state transitions) must usually be designed by hand. A maximum likelihood architecture construction algorithm exists for the special case of building profile HMMs from multiple alignments (Durbin *et al.*, 1998). Efforts have been made to develop architecture learning algorithms for general HMMs (Yada *et al.*, 1996). One can also train fully connected HMMs and prune low probability transitions at the end of training (Mamitsuka, 1996).

More or less formal probabilistic models are increasingly important in biological analysis, particularly in complicated analysis problems with many model parameters. Because many problems in computational biology reduce to some sort of linear "sequence" analysis, probabilistic models based on HMMs have been applied to many problems. Other biological applications of HMMs include genefinding (Burge & Karlin, 1997; Henderson *et al.*, 1997; Krogh *et al.*, 1994b; Kulp *et al.*, 1996; Krogh, 1997; Lukashin & Borodovsky, 1998), radiation hybrid mapping (Slonim *et al.*, 1997), genetic linkage mapping (Kruglyak *et al.*, 1996), phylogenetic analysis (Felsenstein & Churchill, 1996; Thorne *et al.*, 1996), and protein secondary structure prediction (Asai *et al.*, 1993; Goldman *et al.*, 1996). In general, the more a problem resembles a linear sequence analysis problem – that is, the less it depends on correlations between "observables" (e.g. residues) – the more useful HMM approaches will be. Profile HMMs and HMM-based genefinders have probably been the most successful applications of HMMs in computational biology. On the other hand, protein secondary structure prediction is an area in which the state of the art is neural net methods that outperform HMM methods by using extensive local correlation information that is not necessarily easy to model in an HMM (Rost & Sander, 1993).

## Profile HMMs

Krogh *et al.* introduced an HMM architecture that was well suited for representing profiles of multiple sequence alignments (Krogh *et al.*, 1994a). For each consensus column of the multiple alignment, a "match" state models the distribution of residues allowed in the column. An "insert" state and "delete" state at each column allow for insertion of one or more residues between that column and the next, or for deleting the consensus residue. Profile HMMs are strongly linear, left-right models, unlike the general HMM case. Figure 2 shows a small profile HMM corresponding to a short multiple sequence alignment.

The probability parameters in a profile HMM are usually converted to additive log-odds scores before aligning and scoring a query sequence (Barrett *et al.*, 1997). The scores for aligning a residue to a profile match state are therefore comparable to the derivation of BLAST or FASTA scores: if the probability of the match state emitting residue $x$ is $p_x$, and the expected background frequency of residue $x$ in the sequence database is $f_x$, the score for residue $x$ at this match state is $\log p_x / f_x$.

*[Figure 2.]*

For other scores, profile HMM treatment diverges from standard sequence alignment scoring. In traditional gapped alignment, an insert of $x$ residues is typically scored with an affine gap penalty, $a + b(x - 1)$, where $a$ is the score for the first residue and $b$ is the score for each subsequent residue in the insertion. In a profile HMM, for an insertion of length $x$ there is a state transition into an insert state which costs $\log t_{MI}$ (where $t_{MI}$ is the state transition probability for moving from the match state to the insert state), $(x-1)$ state transitions for each subsequent insert state that cost $\log t_{II}$, and a state transition for leaving the insert state that costs $\log t_{IM}$. This is akin to the traditional affine gap penalty, with the gap open cost as $a = \log t_{MI} + \log t_{IM}$, and the gap extend cost as $b = \log t_{II}$.

However, in a profile HMM, these gap costs are not arbitrary numbers. This is an example of why probabilistic models have useful and nontrivial optima. Imagine that we were trying to optimize the gap parameters of a model by maximizing the score of the model on a training set of example sequences. In a profile with *ad hoc* gap costs, we could trivially maximize the scores just by setting all gap costs to zero, but the alignments produced by a profile with no gap penalties would be terrible. In the profile HMM, in contrast, the probability of a transition to an insert is linked to the probability of transition to a match and *not* inserting; profile HMMs have a cost for the match state to match state transition that has no counterpart in standard alignment. As we lower the gap cost by raising the transition probability $t_{MI}$ towards 1.0, the probability of the match-match transition $t_{MM}$ drops towards zero, and thus the cost for sequences without an insertion approaches negative infinity. There is therefore a tradeoff point in choosing the state transition probabilities where the cost for the sequences that do have an insertion is balanced against the cost for the sequences that don't.

Additionally, the inserted residues are associated with insert state emission probabilities in the HMM. If these emission probabilities are the same as the background amino acid frequency, then the score of inserted residues is $\log f_x/f_x = 0$. In traditional alignment, inserted residues also have no cost besides the affine gap penalty. The profile HMM formalism forces us to see that this zero cost corresponds to an assumption that unconserved insertions in protein structures have the same residue distribution as proteins in general. However, the assumption is usually wrong. Insertions tend to be seen most often in surface loops of protein structures, and so have a bias towards hydrophilic residues. Profile HMMs can capture this information in the insert state emission distributions.

## Profile HMM software

Several available software packages implement profile HMMs or HMM-like models (Table I). One important difference between these packages is the model architecture they adopt (Figure 3). The philosophical divide is between "profile" models and "motif" models. By "profile" models, I mean models with an insert and delete state associated with each match state, allowing insertion and deletion anywhere in a target sequence. By "motif" models, I mean models dominated by strings of match states (modeling ungapped blocks of sequence consensus) separated by a small number of insert states modeling the spaces between ungapped blocks.

*[Figure 3.]*

SAM (Hughey & Krogh, 1996), HMMER (S.R.E., unpublished), PFTOOLS (Bucher *et al.*, 1996), and HMMpro (Baldi *et al.*, 1994) implement models based at least in part on the original profile HMMs of Krogh et al. (Krogh *et al.*, 1994a). All three packages have augmented that simple model to deal with multiple domains, sequence fragments, and local alignments, as illustrated by the HMMER 2.0 "Plan 7" model architecture in Figure 3. Thus, local versus global alignment is not necessarily intrinsic to the algorithm (as is usually thought, for instance, in the distinction between the global "Needleman/Wunsch" and local "Smith/Waterman" algorithms), but can be dealt with probabilistically as part of the model architecture. Local alignments with respect to the model are allowed by non-zero state transition probabilities from a begin state to internal match states, and from internal match states to an end state (dotted lines in Figure 3). Local alignments with respect to the sequence are allowed by non-zero state transitions on the flanking insert states (shaded in the Plan 7 architecture in Figure 3). More than one hit to the HMM per sequence is allowed by a cycle of nonzero transitions through a third special insert state.

These profile HMMs are rather general, allowing insertions and deletions anywhere in a sequence relative to the consensus model. Intuitively, they should be more sensitive than ungapped models. However, in practice there is a tradeoff between increasing the descriptive power of the model and the difficulty in determining an increasingly large number of free parameters. A complex model is more prone to overfitting the training data and failing to generalize to other sequences. SAM and HMMER use mixture Dirichlet priors on most distributions to help avoid overfitting and to limit the effective number of free parameters (Sjölander *et al.*, 1996). It is possible to even further reduce the effective number of free parameters by adopting hybrid HMM/neural network techniques (Baldi & Chauvin, 1996). Nonetheless, this relatively unconstrained freedom to insert and delete anywhere makes these models somewhat difficult to train from initially unaligned sequences. HMMER and PFTOOLS are used primarily to build database search models from pre-existing alignments, such as those in the Pfam and PROSITE Profiles databases (see below).

PROBE (Neuwald *et al.*, 1997), META-MEME (with its brethren MEME and MAST) (Grundy *et al.*, 1997), and BLOCKS (Henikoff *et al.*, 1998) assume quite different "motif" models. In these models, alignments consist of one or more ungapped blocks, separated by intervening sequences that are assumed to be random (Figure 3). The handling of these gaps in BLOCKS is *ad hoc.* PROBE and META-MEME adopt probabilistic models for the gaps. META-MEME, interestingly, fits its models into HMMER format. The motif models can therefore be viewed as special cases of profile HMMs; indeed, HMMER, SAM, and PFTOOLS have various options for creating motif-like models. The strength here is that by limiting the freedom of the model *a priori*, the HMM training problem is made more tractable. These approaches can be very powerful for discovering conserved motifs in initially unaligned sets of sequences. PROBE, for instance, has been turned loose on a fully automated exercise in identifying domain families in the current protein database starting with single randomly selected query sequences, with impressive results (Neuwald *et al.*, 1997).

GENEWISE is a sophisticated "framesearch" application that can take a HMMER protein model and search it against EST or genomic DNA, allowing for frameshifts, introns, and sequencing errors (Birney & Durbin, 1997).

PSI-BLAST (Altschul *et al.*, 1997) is not an HMM application *per se*, but it uses some principles of full probabilistic modeling to build HMM-like models from multiple alignments. Like the use of PROBE (Neuwald *et al.*, 1997), PSI-BLAST starts from a single query sequence and collects homologous sequences by BLAST search. These homologues are aligned to the query. An HMM-like search model is built from the multiple alignment. The model is searched against the database, new homologues are discovered and added to the alignment, and a new model is built. The process is iterated until no new homologues are discovered. PROBE and PSI-BLAST both illustrate the power of automating iterative profile searches. The remarkable speed of PSI-BLAST also demonstrates that the fast BLAST algorithm can be applied to position specific scoring systems and gapped alignments, and hence to profile HMMs.

With the exception of PSI-BLAST, profile HMM search algorithms are computationally demanding. Fast hardware implementations of Gribskov profile searches (Gribskov *et al.*, 1987) are available from several manufacturers, including Compugen and Time Logic. These systems are currently being revised to accommodate profile HMMs and the existing PROSITE and PFAM HMM libraries. HMM approaches are also readily parallelized (Grundy *et al.*, 1996; Hughey, 1996). Even more esoteric speedups are also possible. For instance, Intel Corporation has made a white paper available on using MMX assembly instructions to parallelize the Viterbi algorithm and get about a two-fold speed increase on Intel hardware (http://developer.intel.com/drg/mmx/AppNotes/AP569.HTM). This could be significant, since some of the WWW-based HMM servers are backed by Intel processor farms running Linux or FreeBSD, such as the ISREC/Prosite INSECT farm (Jongeneel *et al.*, 1998).

## Profile HMM libraries

Profile HMM software is well suited for modeling a particular sequence family of interest and finding additional remote homologues in a sequence database. Suppose instead that I have a query sequence of interest, and I'm interested in whether this sequence contains one or more known domains. This problem arises especially in high-throughput genome sequence analysis, where standard "top hit" BLAST analyses can be confused by proteins with several distinct domains. Now I need to search the single query sequence against a library of profile HMMs, rather than a single profile HMM against a database of sequences. Building a library of profile HMMs in turn requires a large number of multiple alignments of common protein domains. A database of annotated multiple alignments and prebuilt profile HMMs becomes desirable.

Two large collections of annotated profile HMMs are currently available: the Pfam database (Sonnhammer *et al.*, 1997; Sonnhammer *et al.*, 1998) and the PROSITE Profiles database (Bairoch *et al.*, 1997). The PROSITE Profiles database is a supplement to the widely used PROSITE motifs database; for families that cannot be recognized by simple PROSITE motif patterns (regular expressions which either match a sequence or don't), more sensitive profile HMMs are developed. Both databases are available via WWW servers, including on-line analysis servers for submitting protein sequence queries (Table II). A new European Union funded initiative, called Interpro, has established a collaboration among several sites interested in effective protein domain annotation, including the Pfam, PROSITE, and PRINTS development teams as well as the SWISS-PROT/TREMBL team.

The current prerelease of the PROSITE Profiles database contains profiles for 290 protein domains, and the current Pfam 3.0 release contains 806 profiles. There is substantial overlap between the two collections. It is not meaningful to try to estimate how complete these databases are, because the number of protein families in nature is unknown and probably very large. Although there is much discussion of how many protein families there are – the number 1000 is often cited (Chothia, 1992) – such estimates typically make a false assumption that all families have approximately equal numbers of members (Orengo *et al.*, 1994). However, a small number of families (such as protein kinases, G-protein coupled receptors, and immunoglobulin superfamily domains) account for a disproportionate number of sequences. The two databases are therefore seeing diminishing returns as models of less populous families are developed. For example, the 175 models in Pfam 1.0 recognize one or more domains in about 27% of predicted proteins from the *Caenorhabditis elegans* genome project; the 527 models in Pfam 2.0 recognize about 35%; and the 806 models in Pfam 3.0 recognize about 42% (unpublished data). Thus, a roughly five-fold increase in Pfam database size (175 to 806) resulted in only about a 50% increase in the number of sequences recognized with significant scores. On the bright side, the number of *C. elegans* sequences annotated by one or more Pfam models is starting to approach the number that is hit by one or more informative BLAST similarities to the nonredundant sequence database (42% compared to about 55%).

None of the profile servers is mature. Both profile software and profile databases are rapidly improving and changing. In particular, profile databases typically include domain models that other databases may not yet have. Users are well advised to search several domain annotation servers. The Interpro collaboration is expected to be extremely valuable as the various database teams begin actively sharing alignment and annotation data.

# HMMs for fold recognition

Profile HMMs are sometimes viewed as "mere sequence models". However, profile scores can be calculated from structural data instead of sequences, e.g. "3D/1D profiles" (Bowie *et al.*, 1991; Luthy *et al.*, 1992). These structural profile approaches can readily be put into a full probabilistic, HMM-based framework (Stultz *et al.*, 1993; White *et al.*, 1994). Di Francesco and colleagues have used profile HMMs to model secondary structure symbol sequences by modifying the SAM code to emit an alphabet of protein secondary structure symbols, training models on known secondary structures, and aligning these models to secondary structure predictions of new protein sequences (Di Francesco *et al.*, 1997a; Di Francesco *et al.*, 1997b).

The pejorative appellation of "mere sequence models" seems to be applied to HMMs based on a misunderstanding of the central assumption of position-independence in HMMs. Obviously, neighboring three-dimensional structural contacts influence the types of residue that will be accepted at any given position in a protein structure. How can HMMs that explicitly assume position independence hope to be a realistic model of protein structure?

The assumption of position independence only means that when an HMM state scores a residue in a sequence, it does so independently of the rest of *that* sequence's alignment. However, nothing says that the emission probability distribution at that state can't be determined in the first place from complex three dimensional structural knowledge of the training set. If I know that a residue is buried by spatially neighboring hydrophobic residues, and this environment is approximately *constant* among related structures in the protein family, I can build that knowledge into my model. What HMMs cannot deal with efficiently are long-distance correlations between residues, as is seen in RNA structural alignments, where the complementarity of a pair of distant sequence positions is more important than the identity of either position by itself (Durbin *et al.*, 1998). (Short-distance correlation can be built into HMMs without much difficulty; for example, genefinding HMMs typically model the probability of coding hexamers instead of probabilities of single residues.)

Many current fold recognition methods are not cast as HMMs, but instead as sequence/structure "threading" algorithms with relatively *ad hoc* scores. However, any threading scoring system for which a dynamic programming algorithm can be used to find optimal sequence/structure alignments can be recast as a full probabilistic hidden Markov model. This includes "frozen approximation" methods (Godzik *et al.*, 1992), for instance.

The fold recognition section of the CASP (Current Assessment of Structure Prediction) exercise (Moult *et al.*, 1997) is one of the most interesting anecdotal benchmarks of how HMM techniques perform. In CASP, the sequences of protein "prediction targets" whose structures are soon to be solved by crystallography or NMR are made available to computational structure prediction groups. After the structures become available, the success of the fold predictions are evaluated. Ranking the performance of different methods in CASP is difficult and somewhat subjective (Levitt, 1997). Also, there is usually a variable and sometimes substantial degree of expert human interpretation added to the automated methods (Murzin & Bateman, 1997). Nonetheless, CASP has been a lively venue to explore the strengths and weaknesses of fold recognition methods. At CASP2 last year, HMM-based methods were among the techniques used by several of the most successful prediction groups (Murzin & Bateman, 1997; Di Francesco *et al.*, 1997a; Karplus *et al.*, 1997; Levitt, 1997). Indeed, Murzin and Bateman correctly predicted the folds of all six proteins they attempted, using a combination of profile HMMs, secondary structure prediction, and expert knowledge (Murzin & Bateman, 1997).

# Conclusion

The human genome project threatens to overwhelm us in a deluge of raw sequence data. Successful large scale sequence annotation is so difficult that some people almost seem ready to give up on it (Wheelan & Boguski, 1998). The development of robust methods for automated sequence classification and annotation is imperative. Our hope in developing profile HMM methods is that we can provide a second tier of solid, sensitive, statistically based analysis tools that complement current BLAST and FASTA analyses. The combination of powerful new HMM software and large sequence alignment databases of conserved protein domains should help make this hope a reality.

## Acknowledgements

Work on profile HMMs and Pfam in my lab is supported by NIH/NHGRI R01 HG01363, Monsanto, and Eli Lilly. I thank D. States for pointing out the Intel paper on MMX Viterbi implementations; K. Karplus, R. Hughey, and A. Neuwald for providing pre-publication results; and C. Eddy, S. Johnson, my research group, and three anonymous reviewers for their useful criticism of the manuscript. I also thank the many people in the HMM community with whom I have discussed these issues, especially A. Krogh, P. Bucher, A. Neuwald, B. Grundy, G. Mitchison, the other members of the Pfam consortium (the R. Durbin and E. Sonnhammer groups), and the remarkable UC Santa Cruz HMM group.

## References

Altschul, S. F. (1991). Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.* **219**, 555–565.

Altschul, S. F. & Gish, W. (1996). Local alignment statistics. *Meth. Enzymol.* **266**, 460–480.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucl. Acids Res.* **25**, 3389–3402.

Asai, K., Hayamizu, S., & Handa, K. (1993). Prediction of protein secondary structure by the hidden Markov model. *Comput. Applic. Biosci.* **9**, 141–146.

Attwood, T. K., Beck, M. E., Flower, D. R., Scordis, P., & Selley, J. N. (1998). The PRINTS protein fingerprint database in its fifth year. *Nucl. Acids Res.* **26**, 304–308.

Bairoch, A., Bucher, P., & Hofmann, K. (1997). The PROSITE database, its status in 1997. *Nucl. Acids Res.* **25**, 217–221.

Baldi, P. & Brunak, S. (1998). *Bioinformatics: The Machine Learning Approach.* Boston: MIT Press.

Baldi, P. & Chauvin, Y. (1996). Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Comput.* **8**, 1541–1565.

Baldi, P., Chauvin, Y., Hunkapiller, T., & McClure, M. A. (1994). Hidden Markov models of biological primary sequence information. *Proc. Natl. Acad. Sci. USA,* **91**, 1059–1063.

Barrett, C., Hughey, R., & Karplus, K. (1997). Scoring hidden Markov models. *Comput. Applic. Biosci.* **13**, 191–199.

Barton, G. J. (1990). Protein multiple sequence alignment and flexible pattern matching. *Meth. Enzymol.* **183**, 403–427.

Birney, E. & Durbin, R. (1997). Dynamite: A flexible code generating language for dynamic programming methods used in sequence comparison. *Proc. Fifth Int. Conf. on Intelligent Systems in Molecular Biology,* **5**, 56–64.

Bork, P. & Gibson, T. J. (1996). Applying motif and profile searches. *Meth. Enzymol.* **266**, 162–184.

Bowie, J. U., Luthy, R., & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science,* **253**, 164–170.

Bucher, P., Karplus, K., Moeri, N., & Hofmann, K. (1996). A flexible motif search technique based on generalized profiles. *Comput. Chem.* **20**, 3–23.

Burge, C. & Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268**, 78–94.

Chothia, C. (1992). One thousand families for the molecular biologist. *Nature,* **357**, 543–544.

Churchill, G. A. (1989). Stochastic models for heterogeneous DNA sequences. *Bull. Math. Biol.* **51**, 79–94.

Corpet, F., Gouzy, J., & Kahn, D. (1998). The ProDom database of protein domain families. *Nucl. Acids Res.* **26**, 323–326.

Di Francesco, V., Garnier, J., & Munson, P. J. (1997a). Protein topology recognition from secondary structure sequences: Application of the hidden Markov models to the alpha class proteins. *J. Mol. Biol.* **267**, 446–463.

Di Francesco, V., Geetha, V., Garnier, J., & Munson, P. J. (1997b). Fold recognition using predicted secondary structure sequences and hidden Markov models of protein folds. *Proteins,* **1 (Suppl.)**, 123–128.

Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. J. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge UK: Cambridge University Press.

Eddy, S. R. (1996). Hidden Markov models. *Curr. Opin. Struct. Biol.* **6**, 361–365.

Fabian, P., Murvai, J., Vlahovicek, K., Hegyi, H., & Pongor, S. (1997). The SBASE protein domain library, release 5.0: A collection of annotated protein sequence segments. *Nucl. Acids Res.* **25**, 240–243.

Felsenstein, J. & Churchill, G. (1996). A hidden Markov model approach to variation among sites in rate of evolution. *Mol. Bio. Evol.* **13**, 93–104.

Godzik, A., Kolinski, A., & Skolnick, J. (1992). Topology fingerprint approach to the inverse protein folding problem. *J. Mol. Biol.* **227**, 227–238.

Goldman, N., Thorne, J. L., & Jones, D. T. (1996). Using evolutionary trees in protein secondary structure prediction and other comparative sequence analyses. *J. Mol. Biol.* **263**, 196–208.

Gribskov, M., McLachlan, A. D., & Eisenberg, D. (1987). Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA,* **84**, 4355–4358.

Grundy, W. N., Bailey, T. L., & Elkan, C. P. (1996). ParaMEME: A parallel implementation and a web interface for a DNA and protein motif discovery tool. *Comput. Applic. Biosci.* **12**, 303–310.

Grundy, W. N., Bailey, T. L., Elkan, C. P., & Baker, M. E. (1997). Meta-MEME: Motif-based hidden Markov models of protein families. *Comput. Applic. Biosci.* **13**, 397–406.

Henderson, J., Salzberg, S., & Fasman, K. (1997). Finding genes in human DNA with a hidden Markov model. *J. Comput. Biol.* **4**, 127–141.

Henikoff, S. (1996). Scores for sequence searches and alignments. *Curr. Opin. Struct. Biol.* **6**, 353–360.

Henikoff, S., Greene, E. A., Pietrokovski, S., Bork, P., Attwood, T. K., & Hood, L. (1997). Gene families: The taxonomy of protein paralogs and chimeras. *Science,* **278**, 609–614.

Henikoff, S. & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA,* **89**, 10915–10919.

Henikoff, S., Pietrokovski, S., & Henikoff, J. G. (1998). Superior performance in protein homology detection with the Blocks database servers. *Nucl. Acids Res.* **26**, 309–312.

Hughey, R. (1996). Parallel hardware for sequence comparison and alignment. *Comput. Applic. Biosci.* **12**, 473–479.

Hughey, R. & Krogh, A. (1996). Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *Comput. Applic. Biosci.* **12**, 95–107.

Jaynes, E. T. (1998). *Probability Theory: The Logic of Science.* Available from http://bayes.wustl.edu.

Jongeneel, V., Junier, T., Iseli, C., Hofmann, K., & Bucher, P. (1998). INSECT and MOLLUSCS - supercomputing on the cheap. Available from http:// cmpteam4.unil.ch/biocomputing/mollusc/ INSECT_and_MOLLUSCS.html.

Karchin, R. & Hughey, R. (1998). Weighting hidden Markov models for maximum discrimination. Bioinformatics, in press.

Karplus, K., Sjolander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R., Holm, L., & Sander, C. (1997). Predicting protein structure using hidden Markov models. *Proteins,* **1 (Suppl.)**, 134–139.

Krogh, A. (1997). Two methods for improving performance of an HMM and their application for gene finding. *Proc. Fifth Int. Conf. on Intelligent Systems in Molecular Biology,* **5**, 179–186.

Krogh, A. (1998). An introduction to hidden Markov models for biological sequences. In: *Computational Methods in Molecular Biology,* (Salzberg, S., Searls, D., & Kasif, S., eds) pp. 45–63. Elsevier.

Krogh, A., Brown, M., Mian, I. S., Sjolander, K., & Haussler, D. (1994a). Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.* **235**, 1501–1531.

Krogh, A., Mian, I. S., & Haussler, D. (1994b). A hidden Markov model that finds genes in *E. coli* DNA. *Nucl. Acids Res.* **22**, 4768–4778.

Kruglyak, L., Daly, M. J., Reeve-Daly, M. P., & Lander, E. S. (1996). Parametric and nonparametric linkage analysis: A unified multipoint approach. *Am. J. Hum. Genet.* **58**, 1347–1363.

Kulp, D., Haussler, D., Reese, M. G., & Eeckman, F. H. (1996). A generalized hidden Markov model for the recognition of human genes in DNA. *Proc. Fourth Int. Conf. on Intelligent Systems in Molecular Biology,* **4**, 134–141.

Levitt, M. (1997). Competitive assessment of protein fold recognition and alignment accuracy. *Proteins,* **1 (Suppl.)**, 92–104.

Lukashin, A. V. & Borodovsky, M. (1998). GeneMark.hmm: New solutions for gene finding. *Nucl. Acids Res.* **26**, 1107–1115.

Luthy, R., Bowie, J. U., & Eisenberg, D. (1992). Assessment of protein models with three-dimensional profiles. *Nature,* **356**, 83–85.

Mamitsuka, H. (1996). A learning method of hidden Markov models for sequence discrimination. *J. Comput. Biol.* **3**, 361–373.

Moult, J., Hubbard, T., Bryant, S. H., Fidelis, K., & Pedersen, J. T. (1997). Critical assessment of methods of protein structure prediction (CASP): Round II. *Proteins,* **1 (Suppl.)**, 2–6.

Murzin, A. G. & Bateman, A. (1997). Distant homology recognition using structural classification of proteins. *Proteins,* **1 (Suppl.)**, 105–112.

Neuwald, A. F., Liu, J. S., Lipman, D. J., & Lawrence, C. E. (1997). Extracting protein alignment models from the sequence database. *Nucl. Acids Res.* **25**, 1665–1677.

Orengo, C., Jones, D. T., & Thornton, J. M. (1994). Protein superfamilies and domain superfolds. *Nature,* **372**, 631–634.

Pearson, W. & Lipman, D. (1988). Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA,* **85**, 2444–2448.

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

Rost, B. & Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* **232**, 584–599.

Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I. S., & Haussler, D. (1996). Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *Comput. Applic. Biosci.* **12**, 327–345.

Slonim, D., Kruglyak, L., Stein, L., & Lander, E. (1997). Building human genome maps with radiation hybrids. *J. Comput. Biol.* **4**, 487–504.

Sonnhammer, E. L., Eddy, S. R., & Durbin, R. (1997). Pfam: A comprehensive database of protein families based on seed alignments. *Proteins,* **28**, 405–420.

Sonnhammer, E. L. L., Eddy, S. R., Birney, E., Bateman, A., & Durbin, R. (1998). Pfam: Multiple sequence alignments and HMM-profiles of protein domains. *Nucl. Acids Res.* **26**, 320–322.

Stultz, C. M., White, J. V., & Smith, T. F. (1993). Structural analysis based on state-space modeling. *Protein Sci.* **2**, 305–314.

Sunyaev, S. R., Rodchenkov, I. V., Eisenhaber, F., & Kuznetsov, E. N. (1998). Analysis of the position dependent amino acid probabilities and its application to the search for remote homologues. *RECOMB '98,* pp. 258–265.

Tarnas, C. & Hughey, R. (1998). Reduced space hidden Markov model training. Bioinformatics, in press.

Taylor, W. R. (1986). Identification of protein sequence homology by consensus template alignment. *J. Mol. Biol.* **188**, 233–258.

Thorne, J. L., Goldman, N., & Jones, D. T. (1996). Combining protein evolution and secondary structure. *Mol. Bio. Evol.* **13**, 666–673.

Wheelan, S. J. & Boguski, M. S. (1998). Late-night thoughts on the sequence annotation problem. *Genome Res.* **8**, 168–169.

White, J. V., Stultz, C. M., & Smith, T. F. (1994). Protein classification by stochastic modeling and optimal filtering of amino-acid sequences. *Math. Biosci.* **119**, 35–75.

Wu, C. H., Zhao, S., & Chen, H. L. (1996). A protein class database organized with ProSite protein groups and PIR superfamilies. *J. Comput. Biol.* **3**, 547–561.

Yada, T., Ishikawa, M., Tanaka, H., & Asai, K. (1996). Extraction of hidden Markov model representations of signal patterns in DNA sequences. *Pac. Symp. Biocomput.* pp. 686–696.

| Software | URL |
|---|---|
| SAM | http://www.cse.ucsc.edu/research/compbio/sam.html |
| HMMER | http://genome.wustl.edu/eddy/hmmer.html |
| PFTOOLS | http://ulrec3.unil.ch:80/profile/ |
| HMMpro | http://www.netid.com/ |
| GENEWISE | http://www.sanger.ac.uk/Software/Wise2/ |
| PROBE | ftp://ncbi.nlm.nih.gov/pub/neuwald/probe1.0/ |
| META-MEME | http://www.cse.ucsd.edu/users/bgrundy/metameme.1.0.html |
| BLOCKS | http://www.blocks.fhcrc.org/ |
| PSI-BLAST | http://www.ncbi.nlm.nih.gov/BLAST/newblast.html |

**Table I.** Internet sources for obtaining some of the existing profile HMM and HMM-like software packages.

| | |
|---|---|
| **Profile HMM libraries** | |
| Pfam (Sonnhammer *et al.*, 1998) | http://www.sanger.ac.uk/Pfam/ |
| PROSITE profiles (Bairoch *et al.*, 1997) | http://ulrec3.unil.ch/software/PFSCAN_form.html |
| | |
| **HMM-like methods** | |
| BLOCKS (Henikoff *et al.*, 1998) | http://www.blocks.fhcrc.org/ |
| | |
| **Other protein domain family classification servers** | |
| PRINTS (Attwood *et al.*, 1998) | http://www.biochem.ucl.ac.uk/bsm/dbbrowser/PRINTS/ |
| ProClass (Wu *et al.*, 1996) | http://diana.uthct.edu/proclass.html |
| PRODOM (Corpet *et al.*, 1998) | http://www.toulouse.inra.fr/prodom.html |
| SBASE (Fabian *et al.*, 1997) | http://base.icgeb.trieste.it/sbase/ |

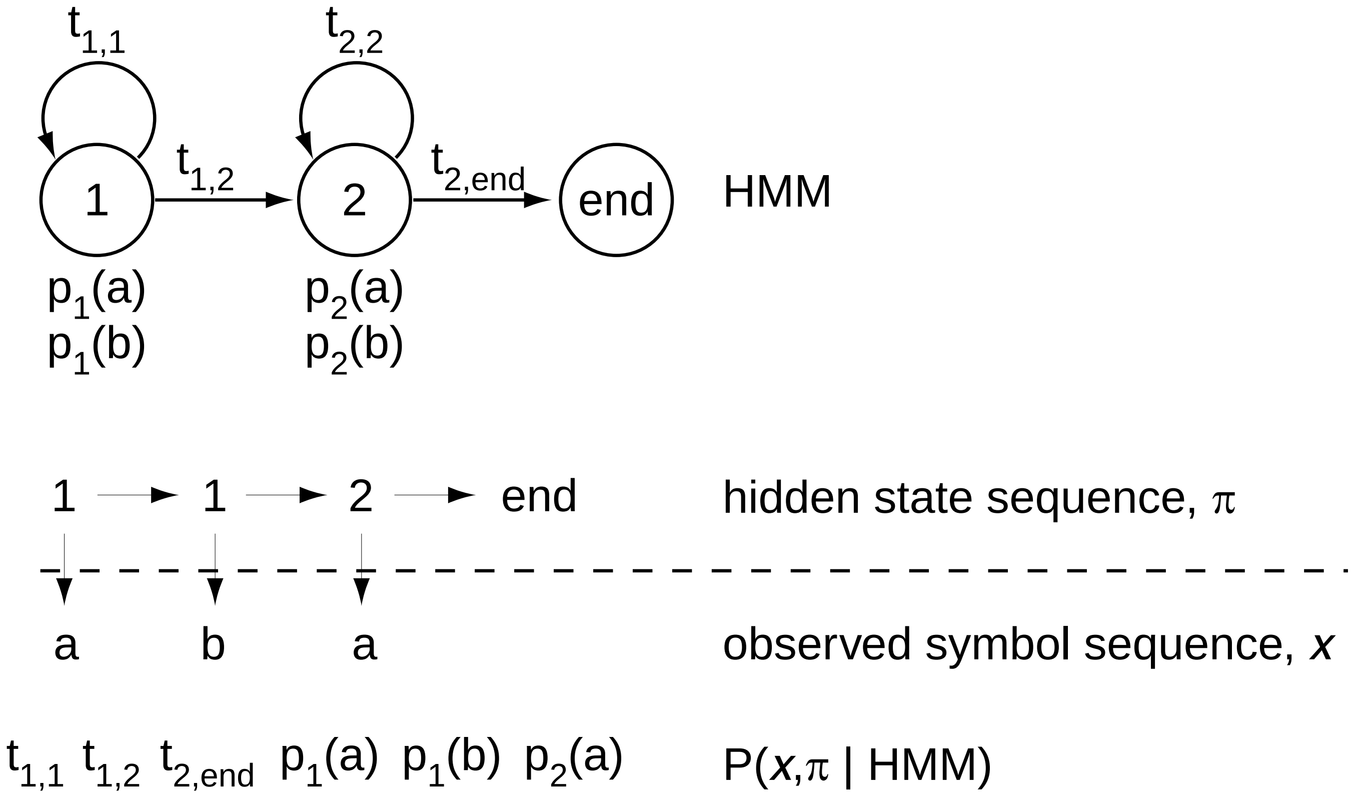
**Table II.** WWW analysis servers for analyzing protein sequences for known domains.

# Figure legends

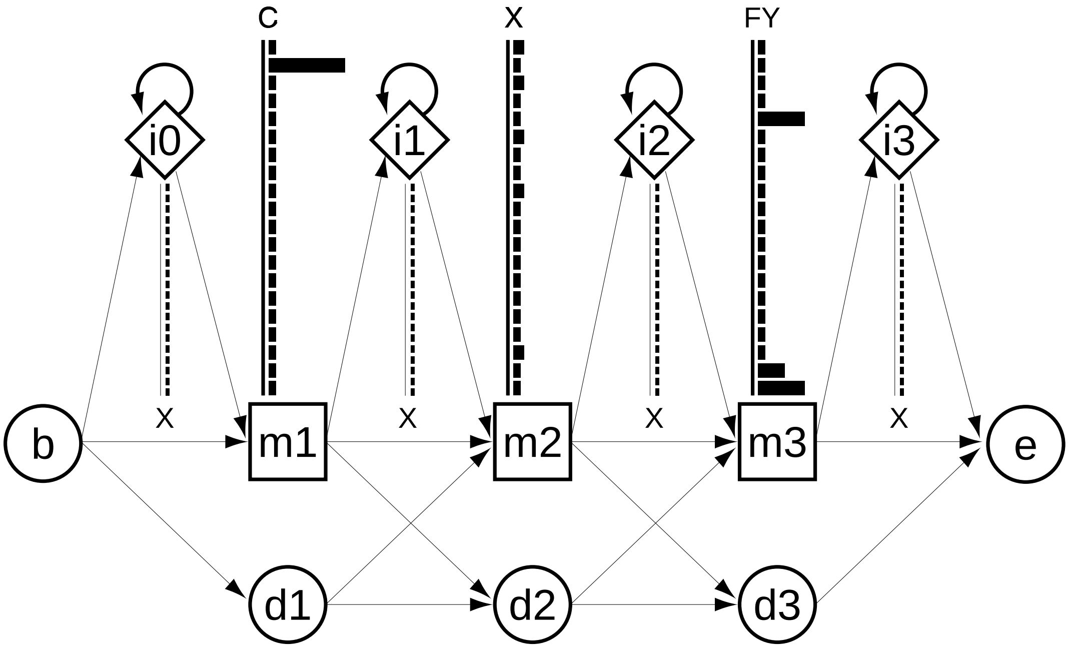
Figure 1. A toy HMM, modeling sequences of a's and b's as two regions of potentially different residue composition. The model is drawn (top) with circles for states, and arrows for state transitions. A possible state sequence generated from the model is shown, followed by a possible symbol sequence. The joint probability $P(x, \pi|\mathrm{HMM})$ of the symbol sequence and the state sequence is a product of all the transition and emission probabilities. Notice that another state sequence (1-2-2) could have generated the same symbol sequence, though probably with a different total probability. This is the distinction between HMMs and a standard Markov model with nothing to hide: in an HMM, the state sequence (e.g. the biologically meaningful alignment) is not uniquely determined by the observed symbol sequence, but must be inferred probabilistically from it.

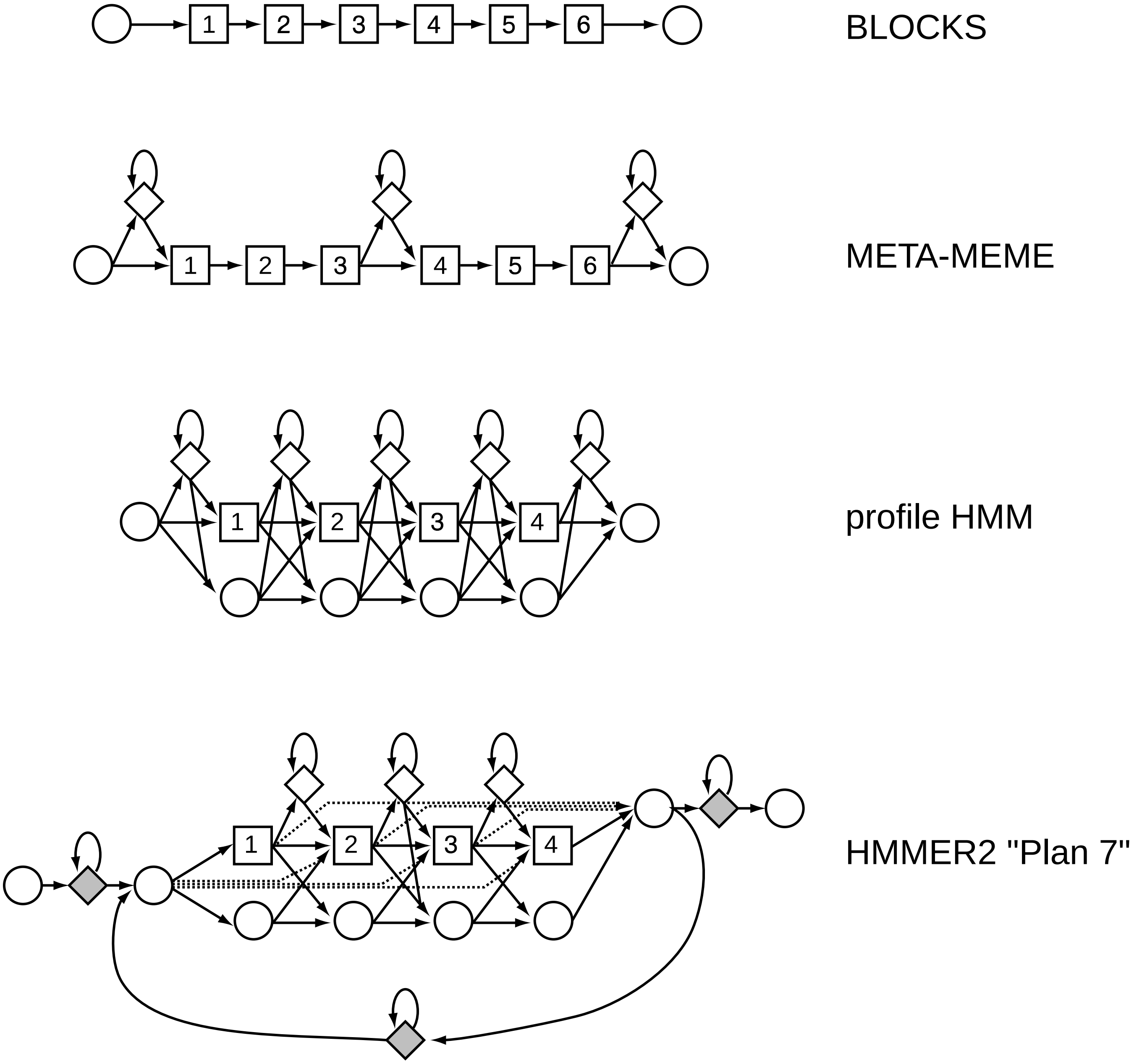
Figure 2. A small profile HMM (right) representing a short multiple alignment of five sequences (left) with three consensus columns. The three columns are modeled by three match states (squares labeled m1, m2, and m3), each of which has 20 residue emission probabilities, shown with black bars. Insert states (diamonds labeled i0 – i3) also have 20 emission probabilities each. Delete states (circles labeled d1 – d3) are "mute" states that have no emission probabilities. A begin and end state are included (b,e). State transition probabilities are shown as arrows.

Figure 3. Different model architectures used in current methods. State transitions are shown as arrows, and emission distributions are not represented. Numbered squares indicate "match states". Diamonds indicate "insert states". Match and insert states each have emission distributions over 4 or 20 possible nucleic or amino acid symbols. Circles indicate nonemitting delete states and other special nonemitting states such as begin and end states. From top to bottom: BLOCKS-style ungapped motifs, represented as an HMM; the multiple motif model in META-MEME; the original profile HMM of Krogh *et al.*; and the "Plan 7" architecture of HMMER 2, representative of the new generation of profile HMM software in SAM, HMMER, and PFTOOLS.


$t_{1,1}$
$t_{2,2}$
$t_{1,2}$
$t_{2,end}$
1
2
end
HMM
$p_1(a)$
$p_1(b)$
$p_2(a)$
$p_2(b)$
1 → 1 → 2 → end
hidden state sequence, $\pi$
a b a
observed symbol sequence, $x$
$t_{1,1}$ $t_{1,2}$ $t_{2,end}$ $p_1(a)$ $p_1(b)$ $p_2(a)$
$P(x,\pi \mid \text{HMM})$

| 1 | 2 | 3 |
|---|---|---|
| C | A | F |
| C | G | W |
| C | D | Y |
| C | V | F |
| C | K | Y |




1
2
3
4
5
6
BLOCKS
1
2
3
4
5
6
META-MEME
1
2
3
4
profile HMM
1
2
3
4
HMMER2 "Plan 7"