

Towards Discovering Structural Signatures of Protein Folds based on Logical Hidden Markov Models

Kristian Kersting¹, T. Raiko^{1,2}, S. Kramer¹, L. De Raedt¹

¹*Institute for Computer Science* ²*Helsinki University of Technology*

Machine Learning Lab

Laboratory of Computer and

University of Freiburg

Information Science,

Georges-Koehler-Allee 079

P.O. Box 5400,

79112 Freiburg, Germany

02015 HUT, Finland

With the growing number of determined protein structures and the availability of classification schemes, it becomes increasingly important to develop computer methods that automatically extract structural signatures for classes of proteins. In this paper, we introduce and apply a new Machine Learning technique, Logical Hidden Markov Models (LOHMMs), to the task of finding structural signatures of folds according to the classification scheme SCOP. Our results indicate that LOHMMs are applicable to this task and possess several advantages over other approaches.

1 Introduction

In recent years, the number of proteins with determined structure has been growing rapidly due to large-scale structural genomics projects. Consequently, the Protein Data Bank (PDB) is growing at high rates. In parallel, researchers have developed classification schemes of proteins based on their sequence, structure and function. The development of classification schemes is a common scientific activity to make sense and gain a deeper understanding of experimental data. Given the determined structures and classification schemes, the discovery of structural characteristics of protein classes becomes an important topic. The primary interest there is to gain insights into structural characteristics of fold classes, but ultimately structural signatures should also be useful for the prediction of protein folds. In fact, some successful approaches in the CASP predictive exercises made use of knowledge about structural signatures. So far, most signatures have been discovered by human experts based on extensive manual/visual inspection of the data. However, few experts in the world are in a position to find/provide these signatures, and few systematic attempts exist to catalog known signatures. So, there is a need to develop computer methods that automatically extract structural signatures in a systematic way¹¹.

Recently, Hidden Markov Models (HMM) have been used to analyze classes in SCOP⁶. HMMs are among the most widely and successfully used tools

for the analysis of sequence data in bioinformatics. Despite their successes, however, it is well-known that HMMs have a number of weaknesses. One of the major weaknesses is that HMMs handle only flat sequences, i.e. sequences of unstructured symbols. In this paper we will overcome this weakness by introducing Logical Hidden Markov Models (LOHMMs).

This paper is organized as follows. In Section 2, we present the task and the dataset. In Section 3, we introduce LOHMMs. Section 4 describes experiments with LOHMMs for the discovery of structural signatures. Subsequently, we discuss related work and conclude.

2 Task and Dataset

In this section, we describe the task of finding structural signatures of protein folds and the dataset used. The basis of our study is the SCOP (Structural Classification of Proteins) database due to A. Murzin and maintained by the MRC Laboratory of Molecular Biology. Our goal was to find structural characteristics of the domains at the second level of the SCOP hierarchy, i.e., the level of folds. In our study, we focused on alpha and beta proteins (a/b), a class consisting of domains with mainly parallel beta sheets (beta-alpha-beta units). From this class, we chose the five most populated subclasses, that is, folds: TIM beta/alpha-barrel, NAD(P)-binding Rossmann-fold domains, Ribosomal protein L4, glucosamine 6-phosphate deaminase/isomerase and leucine aminopeptidase. The overall set-up is quite similar to the one in ^{12,11}. The data have been extracted automatically from the PDB release #96 and SCOP version 1.57.

Information for domains from the above five folds was extracted in the form of “logical sequences” of secondary structure elements. Logical sequences are sequences of logical atoms. An example of such a sequence (corresponding to a Ribosomal protein L4) is:

$$\begin{aligned}
 &st(\textit{null}, 2), he(h(\textit{right}, \textit{alpha}), 6), st(\textit{plus}, 2), he(h(\textit{right}, \textit{alpha}), 4), st(\textit{plus}, 2), \\
 &\quad he(h(\textit{right}, \textit{alpha}), 4), st(\textit{plus}, 3), he(h(\textit{right}, \textit{alpha}), 4), \\
 &\quad st(\textit{plus}, 1), he(h(\textit{right}, \textit{alpha}), 6).
 \end{aligned}$$

There are two predicates *he* and *st*. Atoms *he(Type, Length)* model helices of a certain type and length, whereas atoms *st(Orientation, Length)* model strands of a certain orientation and length. The helix types are: *h(left, alpha)*, *h(right, alpha)*, *h(left, gamma)*, *h(right, gamma)*, *h(left, omega)*, *h(right, omega)*, *h(right, pi)*, *h(right, 3to10)*, *27ribbon* and *polyproline*. The orientation of strands can be *null* (the beginning of a sheet), *plus* (a parallel strand of a sheet), or *minus* (an anti-parallel strand of

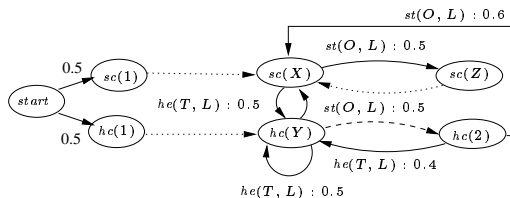


Figure 1: A logical hidden Markov model encoding default reasoning. The dashed edge represents a *more general than* relation.

a sheet). The length is defined as the number of acids and was quantized in the experiments (see below).

For each of the above five folds, we are modelling their domains in terms of their secondary structure using a logical variant of HMMs. So, what can be expected from an application of LOHMMs to this task? First of all, it should be clear that we do not obtain structural signatures for each of these classes immediately. What we obtain instead, is a model for each fold. Each model provides a precise probabilistic and logical characterization of the respective fold. Structural signatures can then be found by a comparison of models. As will be shown below, it is quite easy to find characteristics upon inspection of the trained models.

3 Logical (Hidden) Markov Models

Logical (hidden) Markov models (LOHMM) extend the unstructured model representation of HMMs^{10,9} by incorporating complex, internal structure into the specification of transitions (and therefore of emissions) between states.

Sets of states are summarized by *abstract states*, which are represented by logical atoms. A logical atom then represents all states that can be obtained by instantiating the atoms (i.e. by replacing the variables by terms). E.g. the abstract state $hc(X)$, where X is a variable, could represent the set of states $\{hc(1), hc(2)\}$ depending on the terms (1 and 2 in this case) in the LOHMM. If the logical atom does not contain any variables such as $hc(1)$, it represents a singleton set. Abstract states are connected by *abstract transitions*, which summarize sets of transitions between states. When a transition is made, a state is sampled from the encompassing abstract state. Subsequently an observation symbol is generated in the same manner. We will explain these concepts on an example. For more details, we refer to⁷.

3.1 An Example of a LOHMM

Fig. 1 shows an example of a LOHMM. The vertices in the model represent abstract (hidden) states where the predicate $hc(ID)$ (resp. $sc(ID)$) represents a block ID of consecutive helices (resp. strands). In such models, we find three different types of edges:

Solid edges between abstract states specify the abstract transitions. Transition probabilities and emission symbols are associated to them. An example transition from Fig. 1 is $sc(X) \xrightarrow{st(O,L):0.5} hc(Y)$. Such a solid edge expresses that if one is in one of the states represented by $hc(Y)$ one will go to one of the states in $sc(X)$ with probability 0.5 while emitting a symbol in $st(O, L)$.

Dotted edges indicate that two abstract states behave in exactly the same way. If we follow a transition to an abstract state with an outgoing dotted edge, we will automatically follow that edge. Consider the dotted edge going from $sc(Z)$ to $sc(X)$ in Fig 1. The two abstract states are identical. The dotted edge is needed in this case because the variables appearing in the abstract states are different. We could not have written this using solid edges alone as the meaning of the solid edge $sc(X) \xrightarrow{st(O,L):0.5} sc(X)$ is different from that of $sc(Z) \xrightarrow{st(O,L):0.5} sc(X)$. Whereas the first transition only allows a transition between the same state say $sc(1)$ (because the X is identical), the second one allows transition between different states such as $sc(1)$ and $sc(2)$. In a logical sense, dotted edges implement a kind of recursion.

Dashed edges represent a kind of default reasoning. This is often used to model exceptions. Consider the dashed edge in Fig. 1 connecting $hc(Y)$ and $hc(2)$. This dashed edge denotes that $hc(2)$ is a more specific state than $hc(Y)$. This implies that the set of states represented by the more specific (abstract) state is a subset of that represented by the more general one. Logically speaking, the more specific state $hc(2)$ can be obtained by substituting Y by 2 in the more general state $hc(Y)$. Dashed edges and default reasoning are useful because they represent exceptions. Indeed, in our current example, the outgoing probability labels associated to $hc(2)$ are different from those for $hc(Y)$. This actually implies that the $hc(2)$ acts as an exception to the states represented by $hc(Y)$. So for $Y = 2$ we employ the transitions from $hc(2)$ and for $Y \neq 2$ we follow those indicated by $hc(Y)$.

Let us now explain how the model in Fig. 1 generates the sequence of observations $he(h(right, 3to10), 10)$, $st(plus, 10)$, $st(plus, 15)$, $he(h(right, alpha), 9)$, (cf. Fig. 2). Starting from the artificial state *start*, it chooses an initial abstract state, say $hc(1)$. Forced to follow the dotted edge, it enters the abstract state $hc(Y)$. In each abstract state, the model samples values for all variables that

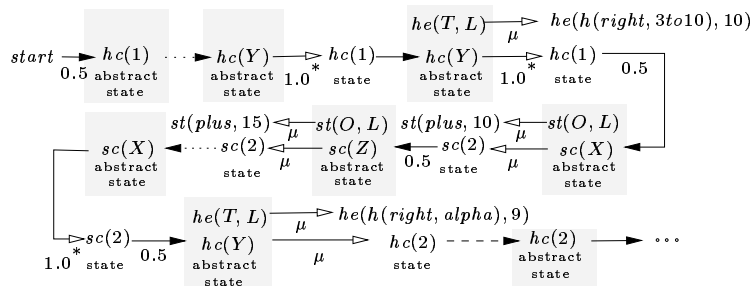


Figure 2: Generating the observation sequence $he(h(right, 3to10), 10)$, $st(plus, 10)$, $st(plus, 15)$, $he(h(right, alpha), 9)$ by the LOHMM in Fig. 1 (* $\mu = 1.0$ due to unification).

are not instantiated yet according to a *selection distribution* μ .

The function μ specifies for each abstract state a distribution over the possible instantiations of the abstract state. E.g. $\mu(he(h(right, alpha), 4) | he(h(T, alpha), 4)) = 0.5$ says that the model samples $he(h(right, alpha), 4)$ with probability 0.5 from $he(h(T, alpha), 4)$ whereas $\mu(he(h(right, alpha), 4) | he(h(T, A), 4)) = 0.05$ specifies that $he(h(right, alpha), 4)$ is sampled with probability 0.05 from $he(h(T, A), 4)$. In general, any probabilistic representation such as Markov chains or Bayesian networks might be used to represent μ . In our experiments, we followed a naïve Bayes approach, i.e. each argument of a predicate is assumed to be independent of the other arguments. E.g., to compute $\mu(he(h(right, alpha), 4) | he(T, L))$, we compute the product of $P_T(h(right, alpha))$ and $P_L(4)$.

Since the value of Y was already instantiated in the previous abstract state $hc(1)$, the model samples with probability 1.0 the state $hc(1)$. It selects the transition to $hc(Y)$ observing $he(T, L)$. Since Y is shared among the head and the body, the state $hc(1)$ is selected with probability 1.0. The observation $he(h(right, 3to10), 10)$ is sampled from $he(T, L)$ using the selection distribution μ . Now, the model goes over to the abstract state $sc(X)$, emitting $st(plus, 10)$ which in turn was sampled from $st(O, L)$. Variable X in $sc(X)$ is not yet bound; so, a value, say 2, is sampled using μ . Next, we move on to abstract state $sc(Z)$, emitting $st(plus, 15)$. The variable Z is sampled to be 3. The dotted edge brings us back to $sc(X)$ and automatically unifies X with Z , which is bound to 3. Emitting $he(h(right, alpha), 9)$, the model returns to abstract state $hc(Y)$. Assume that it samples 2 for variable Y , it has to follow the dashed outgoing edge to $hc(2)$, which represents an exception to $hc(Y)$. This process is similar to unrolling dynamic Bayesian networks² and to grounding logic programs⁸.

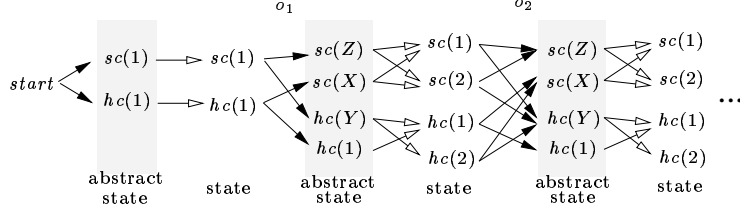


Figure 3: Illustration of the trellis induced by the LOHMM in Fig. 1. In contrast with HMMs, there is an additional layer where the states are sampled from abstract states.

3.2 Semantics and Evaluation

Each HMM is a LOHMM consisting of propositional, logical transitions only. Having the described grounding/unrolling process in mind, it is clear that a LOHMM defines a HMM given a *selection distribution* μ . There are a finite set of abstract transitions and each domain associated to an argument of a predicate is finite. Thus, the set of states and, therefore, the set of (ground) transitions is finite. To summarize, there are two primary differences to HMMs. First, transition probabilities are defined by a product of abstract transition probabilities and the selection probability. Second, the set of states represented by an abstract state can vary with the domains associated to predicates.

A trellis can be built as follows: After selecting an abstract transition, μ generates the relevant states from the head of the abstract transition (cf. Fig. 3). Based on the trellis, it is easy to adapt the *forward-backward*, the *Viterbi* and the *Baum-Welch* algorithms for HMMs to LOHMMs. E.g. in the forward-backward procedure, the probabilities α and β are computed for each reachable state (sets S_t) recursively. The $\alpha_t(s)$ is the probability of the partial observation sequence o_1, \dots, o_{t-1} and state s at time t given the LOHMM. The $\beta_t(s)$ is the probability of the partial observation sequence o_t, \dots, o_T given a state s at time t and the LOHMM. Set $\alpha_0(start) = 1.0$ and $\beta_T(s) = 1.0$ for every $s \in S_T$. Recursive formulae are $\alpha_t(h) = \sum_{cl} \sum_{b \in S_{t-1}} \alpha_{t-1}(b) p_{cl} p_\mu \delta(cl, b, h, o_t)$ and $\beta_t(b) = \sum_{cl} \sum_{h \in S_{t+1}} \beta_{t+1}(h) p_{cl} p_\mu \delta(cl, b, h, o_t)$, where cl is a transition in the LOHMM, p_{cl} is the transition probability and p_μ is the selection probability given by μ . The indicator function $\delta(cl, b, h, o_t) = 1$ whenever transition cl can take from state b to h observing o_t and the transition cl has the most specific body for b . The other algorithms can be adapted analogously.

4 Experiments

The aim of the experiments described below is to put the following hypotheses to test:

- H1** LOHMMs are capable of distinguishing between different folds based on a logical representation of the secondary structure of domains.
- H2** The inspection of LOHMMs reveal distinguishing features of folds.
- H3** LOHMMs can be applied to real-world problems.
- H4** In some applications in computational biology, LOHMMs are by at least an order of magnitude smaller than their instantiations which are HMMs.

We implemented the EM algorithm (with pseudocounts) using the Prolog system Sicstus-3.8.6. The experiments were ran on a Pentium-III-600 MHz machine. Our task was to classify sequences representing protein secondary structures into one of five folds. To do so, we followed the standard approach to classification based on HMMs. We chose a LOHMM (see Fig. 5), fixed its structure, and randomly generated for each fold a set of initial abstract transition probabilities and domain distributions. From each fold dataset ^a described in Section 2, we randomly sampled a training set consisting of 200 sequences. The remaining sequences were used as a test set. Then, we trained these five LOHMMs, one per fold. We used a simple, but common stopping criterion: EM stops if a change in log-likelihood is less than 10^{-1} from one iteration to the next. To evaluate the learned models, we computed the log-likelihood that each model gave to a sequence in the test sets. If the i -th model was the most likely one, then we classified the sequence as a member of class i .

The used LOHMM structure is given in Fig. 5. The hidden states are modelled using $hc(ID, T, L)$ and $sc(ID, O, L)$ representing blocks of consecutive helices and strands. Being in a block ID of consecutive helices (resp. strands), the model will remain in the block or transition to a new block $s(ID)$ of strands (resp. helices). This model takes into account type T , length L and orientation O information. Moreover, there are specific abstract transitions for helices of types $h(right, alpha)$ and $h(right, 3to10)$, and for parallel and anti-parallel strands, and for being at the beginning of a sheet. This enabled us to model the “process” within blocks of consecutive helices quite detailed, and of transitions from blocks of consecutive helices to strands and vice versa. The ID enables

^aFor the extraction of the Prolog facts from the PDB, we adapted the program `secondary.c` made available by the Learning and Planning group of the University of Texas at Arlington (<http://cygnus.uta.edu/subdue/databases/db/proteins.tar.gz>).

Table 1: Confusion matrix showing actual vs. predicted fold classification.

actual \ predicted	fold1	fold2	fold23	fold37	fold55
fold1	736	61	51	62	30
fold2	49	291	53	31	11
fold23	18	23	166	11	15
fold37	55	44	27	282	19
fold55	0	1	1	3	147

Table 2: Precision and recall for each fold rounded to second decimal.

	fold1	fold2	fold23	fold37	fold55
Precision	0.86	0.69	0.56	0.72	0.66
Recall	0.78	0.67	0.71	0.66	0.96

us to have general directed transitions from one block to exactly one successor block.

Results

Our implementation of EM took at most five iterations and approximately 5 minutes to estimate the maximum-likelihood parameters per fold. Given our quantization of the helix and strand lengths, the LOHMM consisted of 74 abstract transition and 46 domain distribution probabilities, whereas the corresponding HMM would consist of over 62,000 transition probabilities. So, the abstract representation of states and transitions in LOHMMs achieves, by design, a remarkable compression of the model, which supports hypothesis H4.

The classification results are summarized by the confusion matrix in Table 1. In this section, the TIM beta/alpha-barrel fold will be denoted as *fold1*, the NAD(P)-binding Rossmann-fold as *fold2*, the Ribosomal protein L4 fold as *fold23*, the glucosamine 6-phosphate deaminase/isomerase fold as *fold37*, and the leucine aminopeptidas fold as *fold55*. In total, 74% (1622 out of 2187) sequences were correctly classified. This result is in the same range as the one reported by ¹¹ (75%). However, we have to emphasize that the datasets are not completely comparable. In contrast to this result, a learner predicting always the majority class would achieve an predictive accuracy of 43%. These results suggest that hypothesis H1 holds. In Table 2, we also give our results in terms of the *recall* and *precision*. Recall is defined as the sum of true positives divided by the sum of true positives and false negatives. Precision is defined as the sum of true positives divided by the sum of true positives and false positives. As can be seen, the recall and precision figures vary among the folds, but within the folds recall and precision are well balanced. In other words, a good

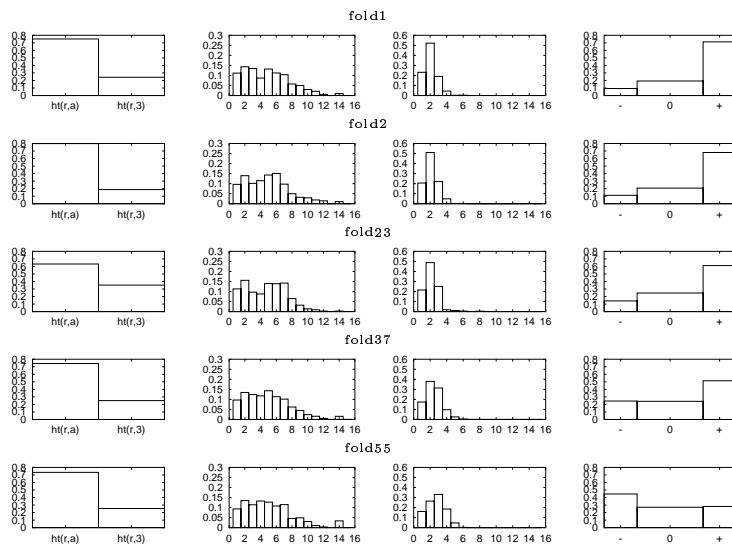


Figure 4: Estimated selection distributions for the five folds (from left to right: helix types, helix lengths, strand lengths, and strand orientations). The distributions specify the probability that a state is sampled from an abstract state (if needed) using a naïve Bayes scheme. The distribution over helix types shows, that only the types $ht(right, alpha)$ (shortly $ht(r, a)$) and $ht(right, 3to10)$ (shortly $ht(r, 3)$) occurred in the data. Due to pseudocounts, no probability value is zero.

precision is not bought at the expense of a good recall, and vice versa. The relatively low precision values for *fold23* and *fold55* are explained by a smaller number of test examples for these two folds. Finally, we inspected the trained LOHMM for characteristic differences. More precisely, we plotted for each of the five estimated LOHMMs the probability distributions implicitly defining μ (see Fig. 4) following a naïve Bayes scheme. Please note that μ defines the probability of sampling a state from an abstract state taking variable binding into account, i.e. that μ and therefore the distributions in Fig. 4 depend on the logical structure of the LOHMM. Upon visual inspection, differences can be found as follows (hypothesis H2):

- Regarding the helix types, *fold23* differs from the others in that the probability of selecting right-handed alpha helices seems to be lower. Also, the probability of right-handed 3to10 helices to be selected seems to be higher than for the other folds.
- According to the strand lengths, we can group the first three and the last

two folds.

- As for strand orientations, we have uniform “patterns” for *fold1*, *fold2* and *fold23*, but characteristic patterns for *fold37* and *fold55*.

To summarize, we believe that the results obtained in our experiments are quite promising also for what concerns the application domain. Therefore, they indicate that the answer to hypothesis H3 should be positive.

5 Related Work

Gough et al.⁶ presented an approach to sequence annotation based on profile HMMs trained on the primary structure of domains for each superfamily in SCOP. The present study is at a different level of abstraction: Firstly, we are working with a secondary structure representation, and not a primary structure representation. Secondly, we are dealing with SCOP folds, not SCOP superfamilies. It might be interesting to apply our approach also at the (more detailed) superfamily level. The goal in⁶ was to annotate sequences based on a library of HMMs that represent all proteins of known structure. In contrast, our short-term goal was to give a proof of the principle, with the intermediate-term goal of providing a tool that helps to gain insights into structural characteristics.

Turcotte et al.^{12,11} applied the Machine Learning and Inductive Logic Programming (ILP) tool Progol to a similar task as the one tackled in this paper. The task there was also to predict SCOP folds based on a high-level logical representation. The difference is that we are working with a larger, more recent dataset, a different representation, and that we are applying a different Machine Learning approach based on probability theory.

HMMs have been extended in a number of different ways e.g. hierarchical HMMs³, factorial HMMs⁵ and based on tree automata⁴. Non of them utilize logical representations. *Relational Markov Models* (RMMs), which were recently introduced and applied to web navigation by Anderson *et. al*¹, are an exception. RMMs do not allow for variable binding, unification nor hidden states.

6 Conclusion

In this paper, we have introduced Logical Hidden Markov Models (LOHMMs) and applied them to the task of finding structural signatures of protein folds. LOHMMs offer the possibility to specify states and transitions at an abstract level, and thereby offer a significant reduction in model size compared to regular HMMs. Our experiments show that the learning performance of LOHMMs is

good. We have also shown that it is easy to extract characteristic patterns from the learned models.

In the future, we will conduct more experiments on more folds in SCOP. Furthermore, we are currently developing algorithms for learning the (logical) structure of LOHMMs.

Acknowledgements This research was partly supported by the European Union IST programme under contract number IST-2001-33053, *APrIL*. T. Raiko was supported by a Marie Curie fellowship at DAISY, HPMT-CT-2001-00251.

1. C. R. Anderson, P. Domingos, and D. S. Weld. Relational Markov Models and their Application to Adaptive Web Navigation. In *Proceedings of KDD-2002*, July 2002.
2. T. Dean and K. Kanazawa. Probabilistic temporal reasoning. In *Proceedings of AAAI-88*, 1988.
3. S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: analysis and applications. *Machine Learning*, 32, 1998.
4. P. Frasconi, G. Soda, and A. Vullo. Hidden markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems*, 18(2/3):195–217, 2002.
5. Z. Ghahramani and M. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.
6. J. Gough, K. Karplus, R. Hughey, and C. Chothia. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. *Journal of Molecular Biology*, 313(4):903–919, 2001.
7. K. Kersting, T. Raiko, S. Kramer, and L. De Raedt. Towards discovering structural signatures of protein folds based on logical hidden markov models. Tech. Rep. 175, University of Freiburg, June 2002.
8. J. W. Lloyd. *Foundations of Logic Programming*. Springer, 1989.
9. L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), 1989.
10. L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, January:4–16, 1986.
11. M. Turcotte, S. Muggleton, and M. J. E. Sternberg. Discovery of structural signatures of protein fold and function. *Journal of Molecular Biology*, 306(3):591–605, 2001.
12. M. Turcotte, S. Muggleton, and M. J. E. Sternberg. The effect of relational background knowledge on learning of protein three-dimensional fold signatures. *Machine Learning*, 43(1/2):81–95, 2001.

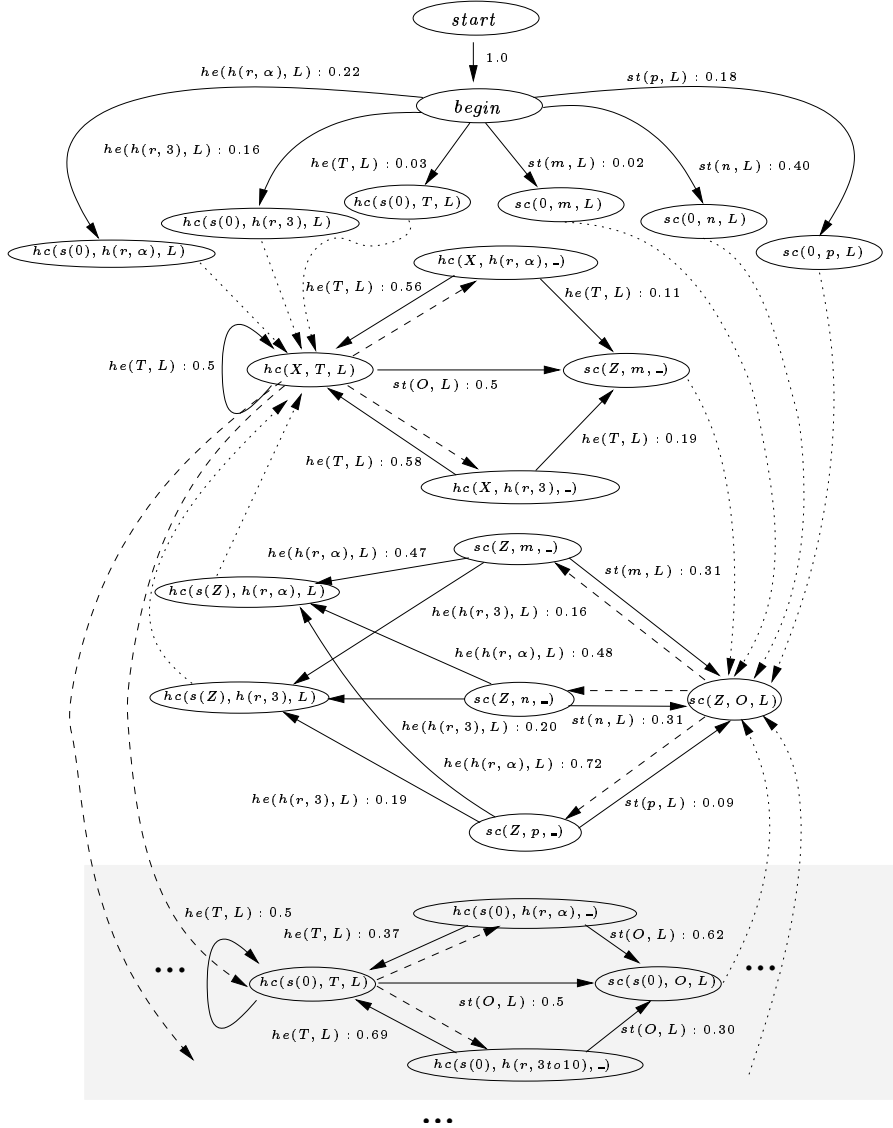


Figure 5: The estimated logical (hidden) Markov model of fold 1. The *end* state is omitted. If probabilities do not sum to 1.0, then there is a transition to *end*. The symbol $_$ denotes anonymous variables which are read and treated as distinct, new variables each time they are encountered. There are copies of the shaded part for $hc(s^2(0), T, L), \dots, hc(s^7(0), T, L)$. Terms are abbreviated using their starting alphanumerical and α for *alpha*.