Homology identification method that combines protein sequence and structure information

Lihua Yu, James White and Temple Smith
1998. Protein Science 7

- OUTLINE
- Introduction
    - Markov Chains
    - Hidden Markov Models (HMMs)
    - Discrete state Space Models (DSMs and pDSMs)
    - Results of experiments described in this paper

From last week

- Shared features of a Protein family (at the sequence level) can be described as a pattern.
- Sequence Pattern and be represented as:
  - Regular Expression (deterministic -> yes/no)
  - Weight Matrix (probabilistic)
  - Profile (probabilistic)
  - HMM (probabilistic)
- Example of a Prosite pattern:
  - [DNSTAGC]-G-D-x(3)-{LIVMF}-G-A
- Example of a profile or a weight matrix:

a	b	a
a	b	-
-	b	a
c	a	-

→

	Col1	Col2	Col3
a	50%	25%	50%
b	0%	75%	0%
c	25%	0%	0%
-	25%	0%	50%

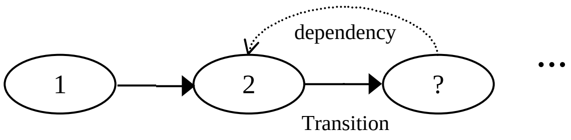
Markov Chains

- Def: A stochastic model for a **series** of random events (such as a time series) whose probabilities at a time interval **depend only** on the previous **K**th event. The series can be a "sequence" of observations over time or space, and the controlling factor is a **transition probability**.

- **Transition probability** is a conditional probability for the system to go to a particular new state, given the Kth previous state of the system.

- Simplest ones are the **first order Markov Chains: K = 1** (model assumption).

-

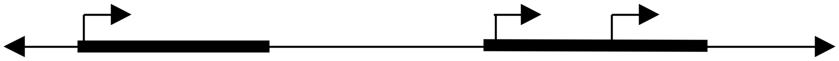


- In the context of biological sequences, can be used to store **primary** structure (raw sequence) and/or higher level structures such as **secondary – quaternary** structure of DNA/RNA/Proteins

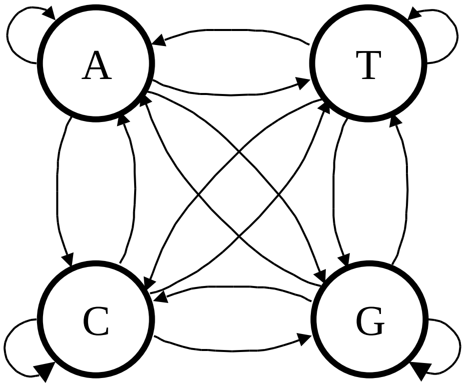
- Simple example from Durbin et al:
    - CpG islands in genomic sequence of H.sapiens and other mammals:
        - In human genome, a 'CpG' pair typically finds it's cytosine has been methylated (chemical modification)
        - Over time, there is high chance that this 'metCpG' will mutate to a 'TG'
          The result is a lower than expected frequency of CpG pairs in the genome (Obs 'CG' < P('C') . P('G'))
        - Evolution has constrained this behaviour to certain areas of genome only. For example, this behaviour is **not** observed around gene promoter regions or inside coding regions.
            - THESE ARE THE CpG ISLANDS !!

CpG island example: M.C

- continues …



- BIOLOGICAL QUESTIONS:
  - 1. Given a sample of genomic sequence, does it come from a CpG island?
  - 2. Given a long piece of sequence, how do we find the CpG islands in it?
- Under a first-order Markov assumption, we want a model in which the probability of a symbol, depends on the previous one, thus we want to model the probability, for example, of finding a “G” given that we already found a “C” symbol. We model all other possibilities as well.
  - Symbols : the alphabet to use: A , G, C, T
  - States: In this case, the same as the symbols (residues)
  - Transitions: Moving from one letter to the next in the sequence
  - Model: A graphical description of the system of states and parameters



If the sequence is:

$X_1, X_2, X_3, \ldots X_L$

The probability of the sequence can be written as follows:

$P(X_1 \ldots X_L) = P(X_L, X_{L-1}, \ldots, X_1)$

Which becomes:

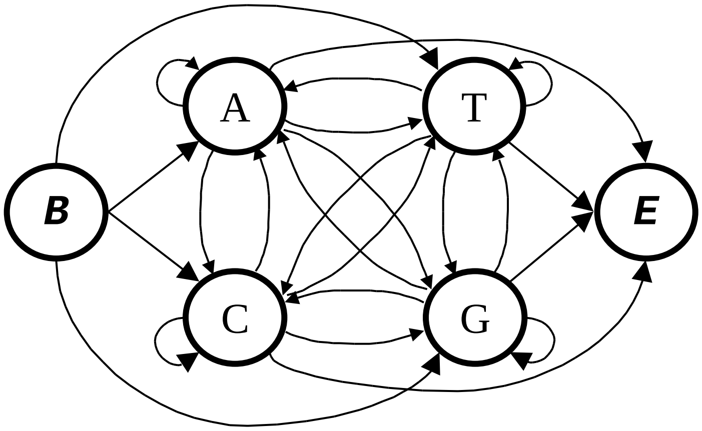
$$P(x) = P(X_L \mid X_1 \ldots X_{L-1}) \, P(X_{L-1} \mid X_1 \ldots X_{L-2}) \ldots . \, P(X_1) ,$$

but first order Markov rule means that

$$P(x) = P(X_L \mid X_{L-1}) \, P(X_{L-1} \mid X_{L-2}) \ldots . \, P(X_2 \mid X_1) \, P(X_1)$$

CpG island example: M.C

- continues …
  Begin and End ‘silent' states can be added to the Markov Chain Model.



- Using a set of real data, two separate MC models can be derived, one for each type of region. The + model is the CpG Island regions, while the – model is the rest of sequence:

+	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.368	**0.274**	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

-	A	C	G	T
A	0.300	0.205	0.285	0.210
C	0.322	0.298	**0.078**	0.302
G	0.248	0.246	0.298	0.208
T	0.177	0.239	0.292	0.292

The transition probabilities were calculated with the equation:

$$a_{st}^{+} = \frac{c_{st}^{+}}{\sum_{t'} c_{st'}^{+}}$$

And its analog for the ‘–’ model, where C^{+}_{st} is the number of times letter t followed letter s in the labeled CpG island regions, the opposite applies for the ‘--’ model . These are the **ML estimators** for the **transition probabilities.**   In the tables, each row sums to 1. Values are for large dataset.

Note G following A is more common than T following A. The CpG effect in the ‘–’ table is obvious as well.

CpG island example: M.C

- continues …
  To answer the first question (discrimination test), calculate the log-odds ratio for sequence *x* of the corresponding transition probabilities.

$$S(x) = \log \frac{P(x \mid Model+)}{P(x \mid Model-)} = \sum_{i=1}^{L} \log \frac{a^{+}_{x_{i-1}x_i}}{a^{-}_{x_{i-1}x_i}}$$

- The following table shows the results of the calculation:

log	A	C	G	T
A	-0.740	0.419	0.580	-0.803
C	-0.913	0.302	1.812	-0.685
G	-0.624	0.461	0.331	-0.730
T	-0.117	0.573	0.393	-0.679

- The authors’ Figure 3.2 shows the distribution of scores S(x) normalized by dividing by their length -> like in average number of bits/molecule

- <SEE FIG 3.2 from DURBIN’s BOOK. P.52>

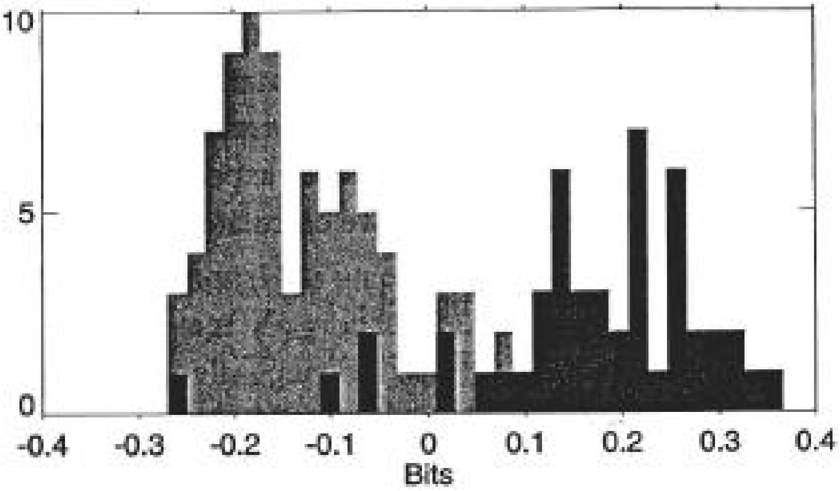


Figure 3.2 *The histogram of the length-normalised scores for all the sequences. CpG islands are shown with dark grey and non-CpG with light grey.*

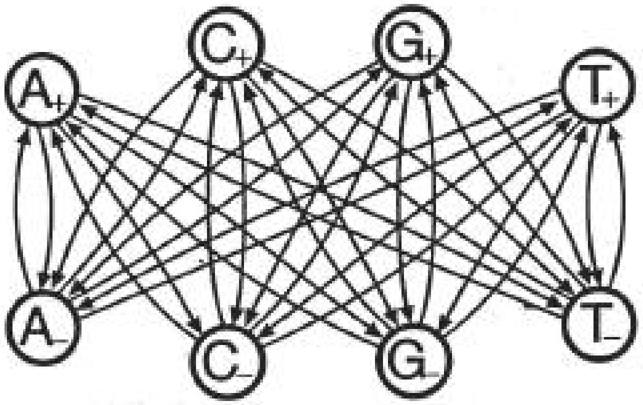
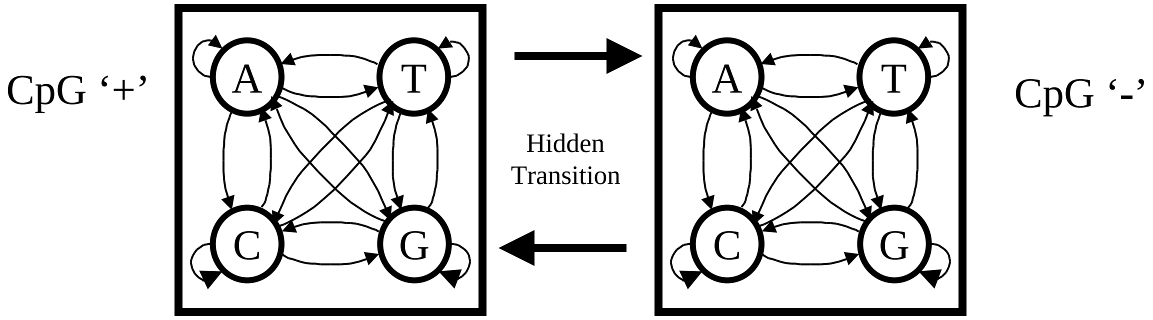


Figure 3.3 *An HMM for CpG islands. In addition to the transitions shown, there is also a complete set of transitions within each set, as in the earlier simple Markov chains.*

To answer the second question:

Hidden Markov Models

- Def: An extension to the M.C. -> Another stochastic generative model. The system randomly evolves from state to state while emitting symbols from the alphabet. When system is at state *i* it has prob. t_{ij} of moving to state *j* and prob. e_{ix} of emitting symbol *X*
  - Symbols : the alphabet to use: A , G, C, T
  - States: State space is discrete (mostly)
  - Transitions: Hidden. Prob Transition matrix (between hidden states)
  - Emissions: Visible. Prob. Emission matrix (between symbols)
  - Model: see in addition figure 7.1 and 7.2 of Brunak et al’s book. p.146



- Only emissions are known (observable), but not the underlying random walk between states, hence the term “hidden”.
- **Differences with M.Cs.**
  - The main difference is the added complexity of the hidden states and the calculation of such state transitions. Hidden states create many possible paths that could generate the observed sequence.
  - In the case of the CpG example, the hidden states are the discrete values “Yes/No” for being in a GpG island at a given time.

A G T G T G C T **C G** A T T G A C A T T **C G** C T **C G** A A T G G T **C G**

Hidden Markov Model

- **General Applications:**
  - First used in speech recognition, later applied in OCR. Also in other fields such as economics and finance.
- **Biological applications:**
  - Modeling of Coding/Non Coding regions, Promoter regions.
  - Modeling of Intron/Exon boundaries
  - Finding protein binding sites in the DNA (i.e. regulation of transcription)
  - Categorization of protein families
  - Multiple alignments
  - Structural analysis and pattern discovery (like above)

- **The main questions to solve**
  - Evaluation (likelihood, discrimination question)
    - Input: the completed model + observed sequence
    - Output: Probability is that the observed sequence was generated by our model.
    - In this calculation, ALL possible PATHS are included (Σ), and an algorithm based on dynamic programming is used to solve: The **Viterbi** algorithm.

    - …

Hidden Markov Model

Continues….

- Decoding
  - Input: the completed model + observed sequence
  - Output: finds most probable path that generated such sequence of states given our model. Equivalent to find the BEST PATH.
  - It also uses the Viterbi algorithm

- Learning (Training question). This is the most difficult of all.
  - Input: A set of sequences (structured data) for training. i.e. The sequences for a Protein Family.
  - Output: Constructs the complete model: Helps designing the general structure (states and connections between them) and obtains the parameters that define such model: transition probabilities and emission probabilities.
  - Several optimising algorithms may be used. The most common is the **EM procedure** (ML type). Others include **Gibbs sampling** (Bayesian solution) and **Gradient descent**

- The Expectation-Maximization (EM) algorithm
  - A type of learning algorithm.
  - begins with an arbitrary set of parameters
  - ML re-estimation of such parameters by considering probable paths for training sequences with the current model. This indicates how they may be modified to improve on the current model
  - try again. The process is iterated until some stopping criterion is reached (like not being able to improve beyond a threshold).

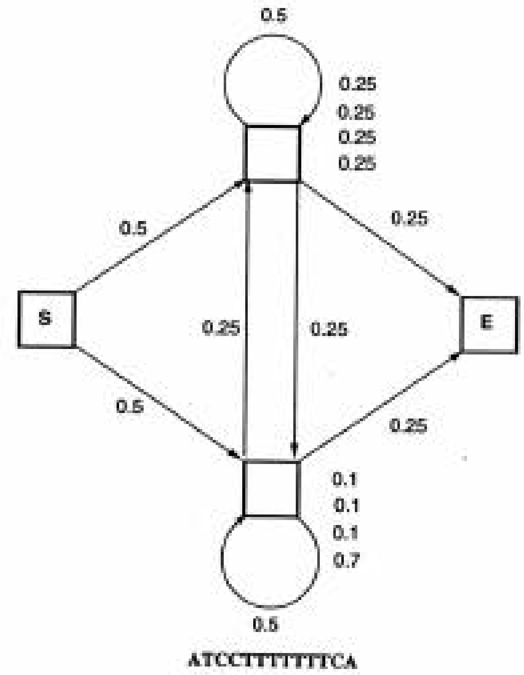


Figure 7.1: A Simple Example of an HMM, with Two States in Addition to the *Start* and *End* States.

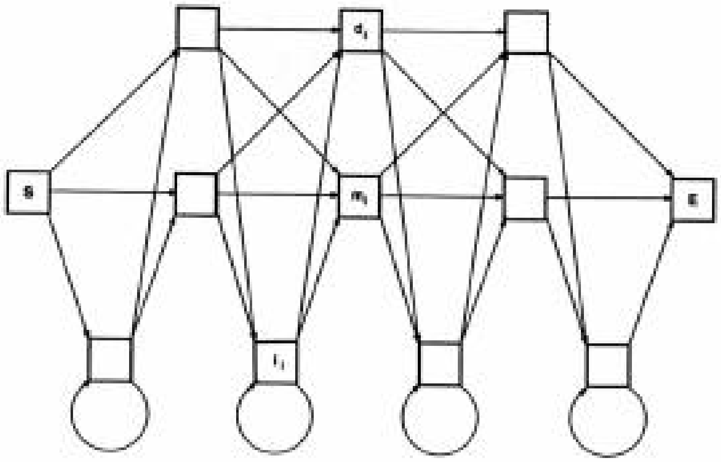


Figure 7.2: The Standard HMM Architecture. S is the start state, E is the end state, and $d_i, m_i,$

Discrete state-space models DSMs

- Examples we saw before included primary structure only, but we can model higher structure information, such as secondary structure.

- A DSM is an idealized representation of a particular tertiary structure class → alpha box, antiparallel bundle, central beta-sheet, barrel, etc.

- The DSMs can be viewed as automatic generators of a.a sequences. They are stochastic.

- Each DSM describes **probabilistically** (Fig 1, 1993 paper)

  - allowed secondary structural elements, types (α-helix , β -strand/ β-sheet, coil/loop/turn) associated with particular folds.

  - Lengths and connectivity (antiparallel, barrel, etc)

  - a.a composition (as well as relative residue positions within the secondary structures and the relative exposure of residues to the solvent)

- All these elements are modeled in a hierarchy of states in a Markov Chain, with transitions between states determined by a transition prob. matrix.

- A number of general protein folds have been modeled with DSMs by the authors (see 1993 paper by same authors) from PDB data.

  → Given a sequence of unknown structure, determine the probability that EACH model has generated it, using a Bayesian filtering algorithm (find posterior probability of each model given the observed sequence)

  → Once the most probable model is found, the most probable secondary structure for each residue is calculated for the sequence (Fig 4)

- Their mathematical structure is **the same** as the one used for **HMMs**

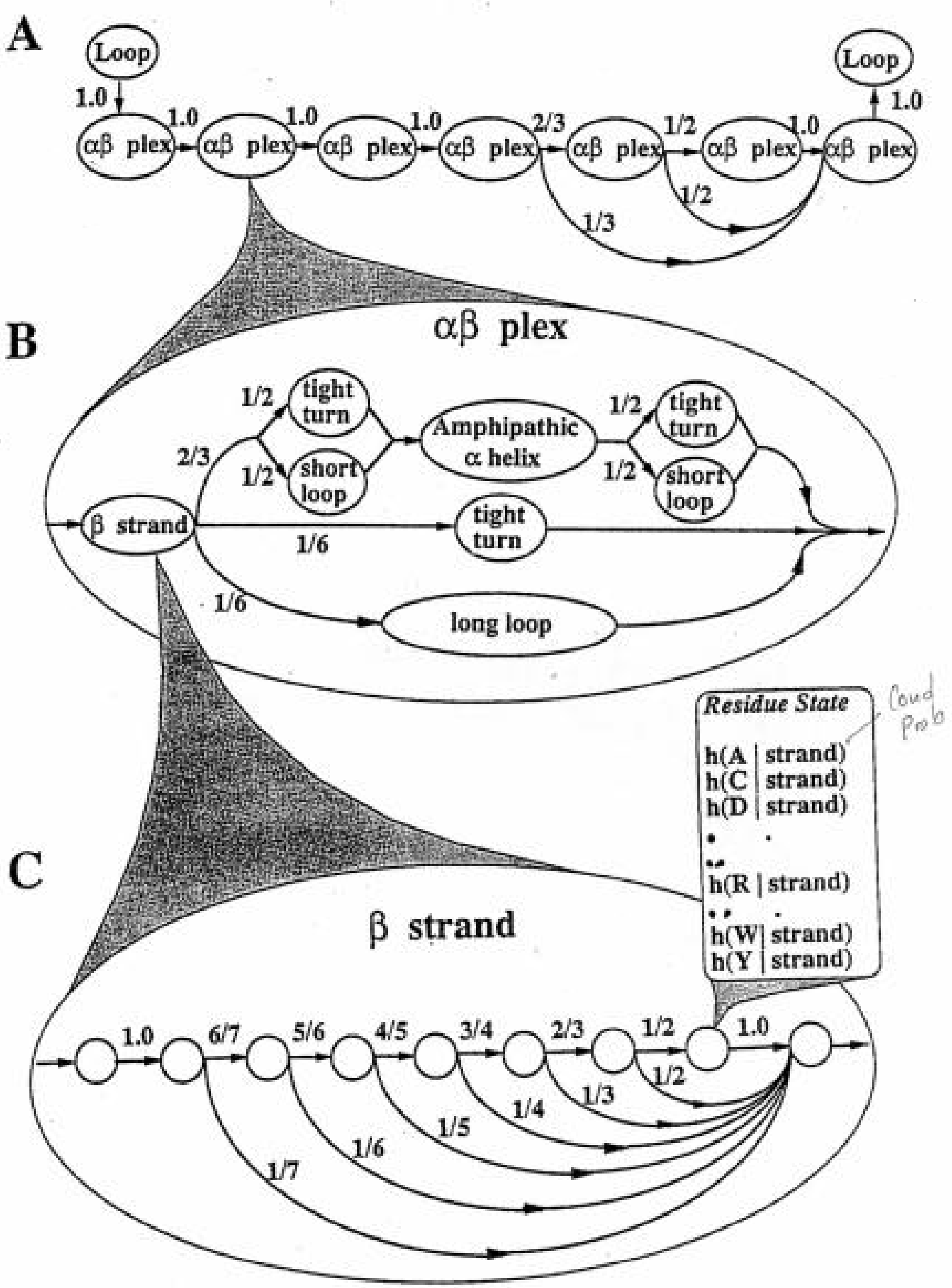


Fig. 1. Schematic of model for the α/β structural class having a central β-sheet containing five, six, or seven β-strands. **A**: The DSM organization at the β plex level of detail. **B**: The β plex is composed of secondary structural elements. **C**: One of these elements, the β-strand submodel, which has a chain of discrete states at the amino acid position level of detail.

Discrete state-space models DSMs

Differences with Hidden Markov Models

- DSMs DO NOT use a training procedure to create the model. Therefore, estimation of transition probabilities is different:
  - Start with a stationary model, based on EXPERT protein knowledge
  - Based on physical interpretation of structural fold, build model that encompasses all possible members (all possible sequences annotated as a given fold type in database of structures).

Modified Discrete state-space models pDSMs

- Starting with a defined DSM for a fold, change the residue probability associated with secondary structural states to a distribution of conserved sequence patterns elements.

- Equivalent to say that functionally conserved sequence patterns are embedded into the model (this is primary structure information).

- The final model combines primary sequence and secondary/tertiary sequence structure. See figure 6

- One advantage is that not training is required. Derived from expert knowledge only (observation of distributions in curated dataset). But this may also be thought of as a disadvantage by others

pDSMs

- The inclusion of conserved sequence patterns assigns zero probabilities to certain states and emissions. While in the case of HMMs, even the very unlikely states have a chance of happening (fig 6).
- The space of possible paths is reduced drastically
- GO TO RESULTS

Limitations of HMM & pDSM

- Limitations of HSMs
  - Often have very large number of parameters to estimate
  - Training of Model is very difficult, and EM algorithm may give sup-optimal answer (falling in local minima region)
  - They are limited by their first order markov property. i.e. They cannot express dependencies between hidden states such as long-range correlations, like certain a.a proximity properties (from 3D folding). Unless these properties are consistently present in the training set.

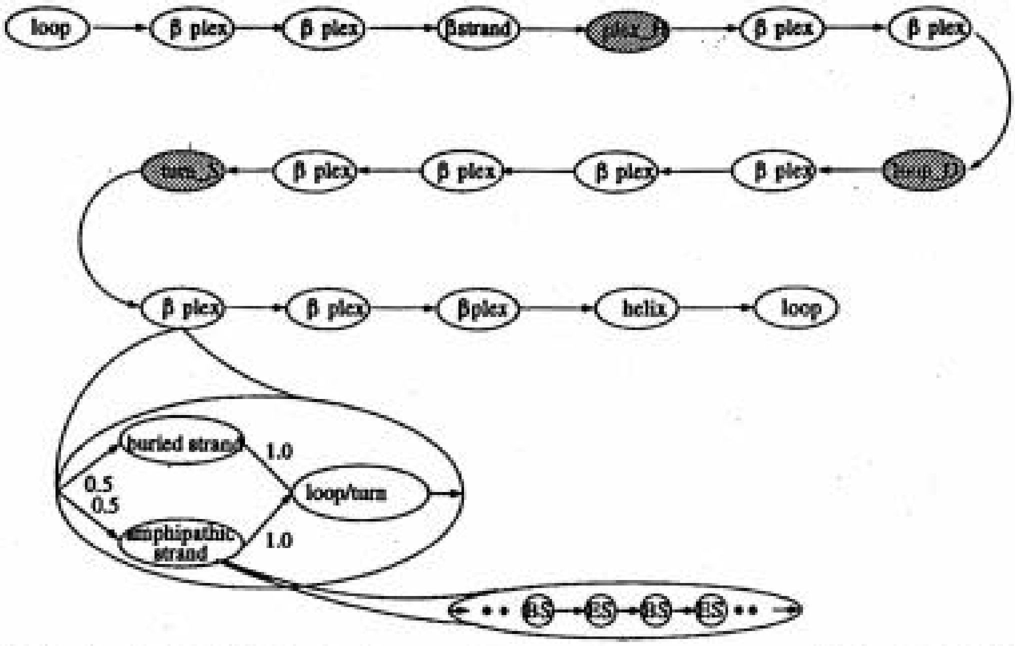


Fig. 5. Schematic of the pDSM for cluster 1 serine proteases. Each oval circle represents a structural building block. Each *β plex* consists of a *β* strand (buried or amphipathic) and the connecting region, which is formed by loops, turns, and sometimes, short helices.

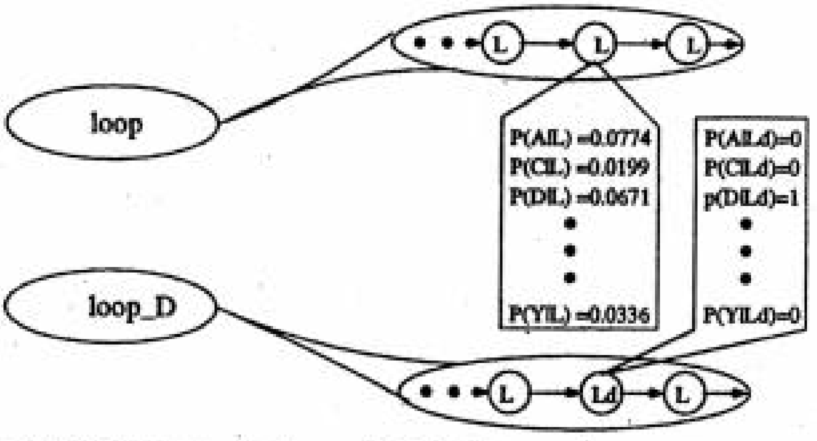
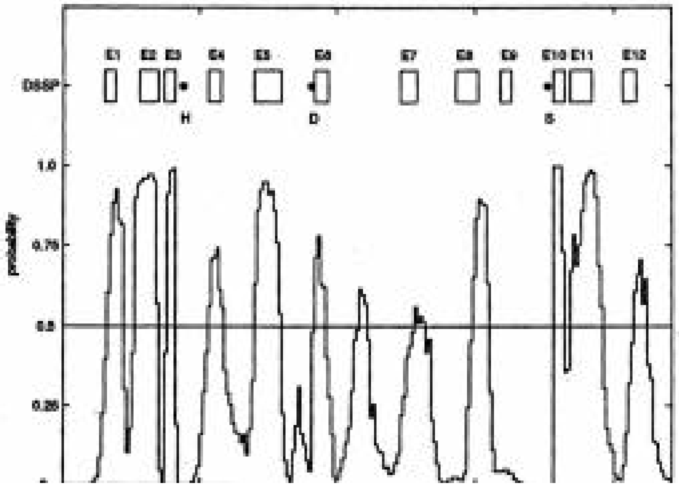


Fig. 6. Illustration of embedding sequence pattern elements into DSMs. The loop structural state (L) is replaced by the conserved sequence pattern element distribution. As shown in the *loop_D* case for the serine proteases, this is a conserved Asp only.



Limitations of HMM & pDSM

Limitations of pDSMs

- Need to construct models "manually" for every protein family. Fig 5 shows one for a cluster within the serine proteases
- Model is not really "optimal", since no EM procedure was used. However, this is the main point the authors of DSMs want to stress as being also a problem in HMMs, and instead they opt for inclusion of expert knowledge

- As with HMMs, there some loss of information in the mapping of 3D information into secondary structure, such as long range residue proximity and contacts
- DOES NOT work with multidomain proteins. The models are made for single domains only. This could be modified.
- Could actually use HMMs to improve further on their already "good" models.

Results of the pDSM paper

Two protein families tested on pDSMs method with a set of false positives and false negatives to test

- Trypsin-like serine proteases (his-Asp-Ser triad)
  - Diverse, with >200 structures available, > 400 sequences. %ID can be as low as 10%.
  - Pattern is X_{24-69} H X_{18-86} D X_{40-109} S X_{44-141} (his-Asp-Ser triad)
- Globins
  - Used a very generic pattern: X_{41-60} F X_{38} H X_{43-68}

Performed a genome scanning of three fully sequence genomes to find new putative members.

Comparisons with other methods : BLAST, Patterns

See results in tables.

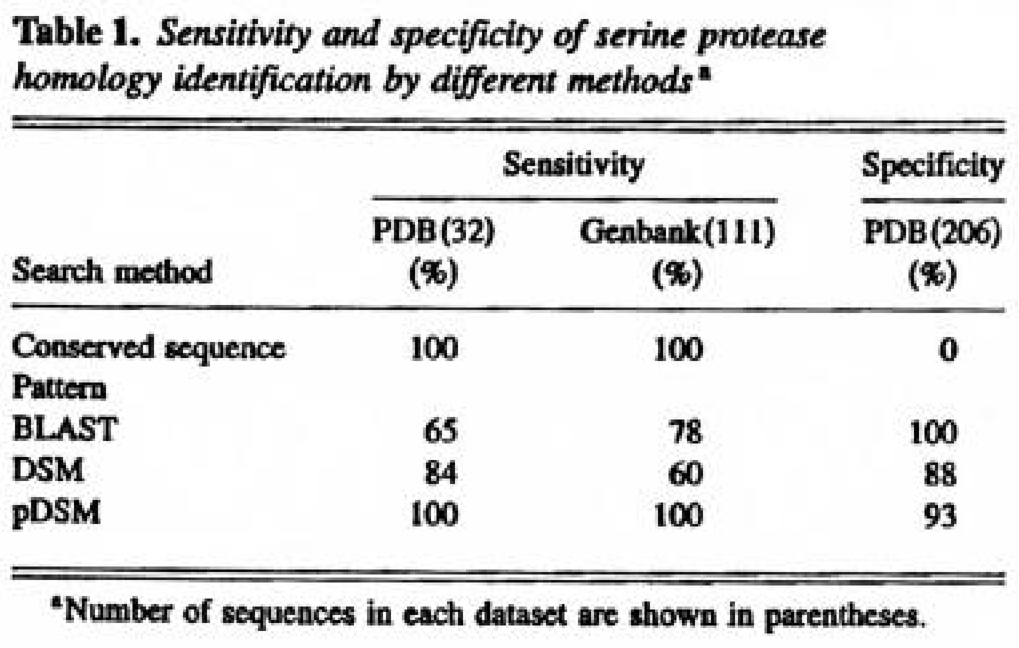


Table 1. *Sensitivity and specificity of serine protease homology identification by different methods*[a]

Search method	Sensitivity		Specificity
	PDB(32) (%)	Genbank(111) (%)	PDB(206) (%)
Conserved sequence Pattern	100	100	0
BLAST	65	78	100
DSM	84	60	88
pDSM	100	100	93

[a] Number of sequences in each dataset are shown in parentheses.

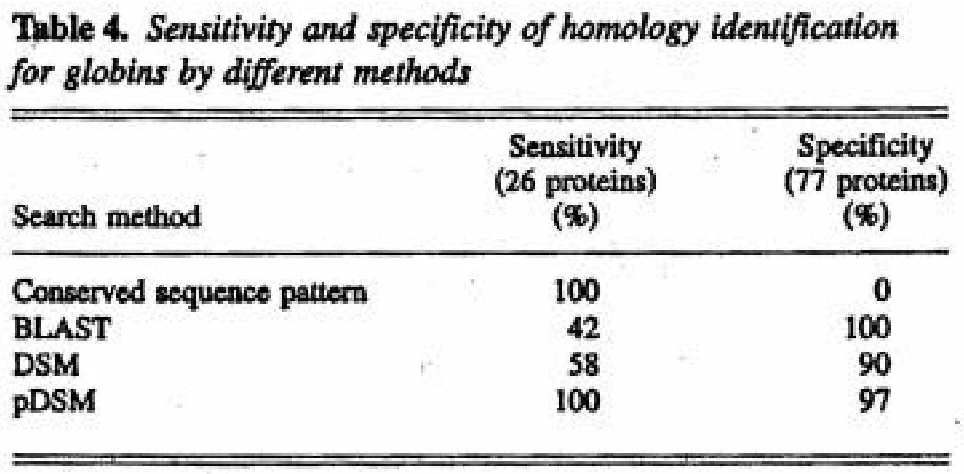


Table 4. *Sensitivity and specificity of homology identification for globins by different methods*

Search method	Sensitivity (26 proteins) (%)	Specificity (77 proteins) (%)
Conserved sequence pattern	100	0
BLAST	42	100
DSM	58	90
pDSM	100	97

Table 2. *The potential trypsin-like serine proteases in genomes identified by pDSM sequence analysis*

	***B. subtilis*: MPR_PBS**
Prediction	(1) Probability: 0.85 (2) Serine protease domain: 104–313 (3) Catalytic triad: His146, Asp191, Ser267
Comment	(1) Annotation[a]: extracellular metalloprotease (Rufo et al., 1996) (2) Weakly similar to 1TRY and 1ELT (PDB), similar to GSEP_BACLI[b] (SWISS-PROT) (3) Signature[c]: TRYPSIN_HIS (4) Alignment with known serine protease: Figure 1
	***E. coli*: b1598**
Prediction	(1) Probability: 0.86 (2) Serine protease domain: entire sequence (3) Catalytic triad: His84, Asp145, Ser223
Comment	(1) Annotation: 24% identical to MPR_BACSU (2) Weakly similar to MPR_PBS and GSEP_BACLI (SWISS-PROT) (3) Signatures: TRYPSIN_HIS and TRYPSIN_SER (4) Alignment with known serine protease: Figure 1
	***S. cerevisiae*: YNL123W**
Prediction	(1) Probability: 0.85 (2) Serine protease domain: 76–286 (3) Catalytic triad: His121, Asp152, Ser236
Comment	(1) Annotation: weak similarity to *C. jejuni* serine protease (2) Similar to HTRA_ECOLI (SWISS-PROT) (3) Signature: none (4) Alignment with known serine protease: Figure 2
	***C. elegans*: CEIV000158**
Prediction	(1) Probability: 0.95 (2) Serine protease domain: entire sequence (3) Catalytic triad: His69, Asp117, Ser212
Comment	(1) Annotation: similar to peptidase family S1 (trypsin) (2) Similar to 1PFX, etc.[d] (3) Signature: TRYPSIN_HIS and TRYPSIN_SER (4) Alignment with known serine proteases: Figure 3

[a] The annotations are obtained from the original genome databases.

[b] GSEP_BACLI has recently been identified as a remote homolog of trypsin-like serine proteases by sequence analysis (Alexandre et al., 1996; Pearson, 1997).

[c] The signatures of serine proteases are defined in PROSITE (Bairoch, 1991).

[d] CEIV000158 matches many serine proteases by BLAST search.

Table 5. *The Q3 accuracy, sensitivity, and specificity of helix prediction for globins by DSMs and pDSMs*[a]

	Q3 (%)		Sensitivity (%)		Specificity (%)	
Loci	DSM	pDSM	DSM	pDSM	DSM	pDSM
1ASH	76	82	89	92	34	51
1BBBA	77	82	91	92	36	53
1BVC	88	87	96	94	64	67
1CMYB	25	77	19	91	92	39
1ECA	47	71	45	90	28	26
1FDHG	55	80	45	91	57	49
1FLP	82	85	95	92	44	67
1FSLA	76	83	91	92	32	59
1GDI	83	85	97	92	41	65
1HBG	78	80	94	92	32	49
1HBHA	51	83	53	94	50	58
1HBHB	61	81	61	93	54	46
1HBIA	51	84	53	92	41	62
1HDSA	58	72	61	100	54	42
1HDSB	43	60	49	88	45	28
1HLB	76	75	99	97	41	43
1HLM	72	72	99	95	39	43
1ITHA	76	81	91	92	40	57
1LHS	86	88	96	96	59	64
1MBA	80	79	91	90	46	49
1MYT	76	82	94	95	34	52
1OUTA	55	82	45	88	56	67
1OUTB	25	79	19	89	88	55
1SCTA	58	79	53	88	60	47
1SCTB	80	84	91	91	37	60
2LHB	80	81	95	94	42	51
Average	66	80(62)	73	92(62)	48	52(74)
STD	18	6	26	3	16	11

[a] The DSSP secondary structure assignments are taken as the true secondary structure. The numbers in parentheses, listed in the row Average, were obtained using the GOR algorithm (Garnier et al., 1978) and are for comparison only. The numbers in row STD are the standard deviations of each column.