Classification of Proteins via Successive State Splitting Algorithm of Hidden Markov Network

Hidetoshi Tanaka¹
ltanaka@icot.or.jp

Kentaro Onizuka¹
onizuka@icot.or.jp

Kiyoshi Asai²
asai@etl.go.jp

¹ Institute for New Generation Computer Technology (ICOT)
1-4-28 Mita, Minato-ku, Tokyo 108

² Electrotechnical Laboratory (ETL)
1-1-4 Umezono, Tsukuba 305

Abstract

Hidden Markov Model (HMM) introduces a stochastic approach to protein representation and motif abstraction. We need the stochastic classification which is seamless with HMM representation and abstraction. Successive State Splitting (SSS) classifies proteins represented by HMM. It uses no previous knowledge of the proteins. The SSS algorithm was originally developed for allophone modeling. It is based on continuous distribution of phenotype data. It enables to obtain an appropriate Hidden Markov Network automatically, and HMM simultaneously. We map amino acids onto continuous space according to quantification based on PAM-250.

1 Introduction

Classification, abstraction, and representation are three significant processes in protein information analysis. We classify proteins by amino acid sequences, secondary or super-secondary structures, 3D structures, and functions. We abstract the consensus patterns from the classified proteins by multiple sequence alignment. And we represent the patterns as motifs[7] or profiles[3]. We propose to use stochastic approach at every process, so that we can obtain better estimation of unknown proteins in their functions and structures.

We consider it most invaluable to classify proteins by their sequences. Proteins are classified by various criteria such as their functions, structures, organizations, and amino acid sequences. The representative classifications are superfamly in PIR[5] for amino acid sequences and helix-strand-turn classification in PDB[4] for secondary structures. Protein sequence classification is used for an index system for similarity searches in databases of amino acid sequences, which are much larger than ones of protein functions, structures,
and their chemical reactions. It enables us to predict functions and structures of unknown proteins from their amino acid sequences rapidly.

The representation for the classified proteins is as important as the classification. We determine a specific consensus pattern (motif), or a variation of protein sequences (profile) through the multiple alignment of their amino acid sequences to represent the classified proteins. The former uses only regular-expression-like patterns while the latter uses their distribution for representing the group of proteins.

To obtain the protein representation, we should abstract it from the amino acid sequences. We have been using multiple sequence alignment for abstraction of the consensus patterns. It makes us understand relationship among protein functions, structures, amino acid sequences. *Functional motifs* are abstracted from the amino acid sequences of similar functions, and *structural motifs* are from the ones of similar structures, so that relationships between functions and structures are explored on the motif basis.

We have studied on *Hidden Markov Model* (HMM)[6] for secondary structure prediction [1] [2], and necessity of private knowledge bases for biological researches[9]. We consider HMM is adequate for the representation of protein classifications as well as abstraction of classified proteins. And we consider *Successive State Splitting* (SSS) algorithm is adequate for protein classification when we employ HMM as their representation.

## 2 SSS Algorithm in Protein Classification

We should emphasize the local similarities of sequences with various scores instead of the similarity of whole sequences with scalar scores such as superfamily classifications. We should introduce stochastic approach to the classification. And we should find global classification by *divisive* approach to obtain even distribution of proteins.

In this Section, we show an algorithm of SSS and compare it with superfamily classification system. It is important to generate a precise and robust HMM: representation capability is necessary for the precision of the model, whereas simplicity is necessary for the robustness of the model. In order to build the robust and precise HMM, it is quite essential to determine an appropriate HMNet.

The SSS algorithm is for building an appropriate HMNet with the *maximum likelihood* criterion. SSS consists of 5 steps: from Step 0 to Step 4.

- **Step 0**: Training of the Initial Model
- **Step 1**: Estimation of the width of PDFs
- **Step 2**: Split of the State
- **Step 3**: Retraining of the Model
- **Step 4**: Change of Distribution

At every step, we explain SSS algorithm for phonemes (allophone modeling), and show the difference between phonemes and amino acids.
Figure 1: SSS Algorithm[8]: Every box indicates the HMNet of respective stages of the SSS algorithm. Each (double) circle represents a state with a self transition and each arc between circles represents a state-transition in the HMNet. (n) correspond to Step n.

2.1 SSS Algorithm

Step 0: Training of the Initial Model

Let the initial model be the HMM of one state, whose output is according to the probability density function (PDF) of the diagonal-covariance 2-mixture Gaussian. The model is trained by all data with a learning algorithm such as forward-backward (Baum-Welch) algorithm or Viterbi algorithm.

As amino acids have discrete distribution while phonemes have continuous one, we should find out common grounds between them to apply SSS algorithm to the protein classification. There are two approaches: to put amino acids into a continuous space (a continuous protein SSS) and to develop a discrete SSS algorithm for proteins. We chose former approach.

Step 1: Estimation of the width of PDFs

We should choose one state to split. For each state $i$, a criterion $d_i$ is calculated by the following equation in the original phoneme SSS.

$$d_i = \sum_k \frac{\sigma_{ik}}{\sigma_{Tk}} \sigma_{ik}^2 = \lambda_{i1} \sigma_{i1k}^2 + \lambda_{i2} \sigma_{i2k}^2 + \lambda_{i1} \lambda_{i2} (\mu_{i1k} - \mu_{i2k})^2$$

(1)

$K$: parameter dimension,
$\lambda_{i1}, \lambda_{i2}$: weight coefficients of mixture Gaussians of S(i),
$\mu_{i1k}, \mu_{i2k}$: $k$-th means of S(i),
$\sigma_{i1k}, \sigma_{i2k}$: $k$-th variances of S(i),
$n_i$: the number of training samples for S(i),
$\sigma_{Tk}$: $k$-th variance of all samples.

It represents a size of the output PDF of 2-mixture Gaussian on the state S(i). It is applicable as it is to the continuous protein SSS.
Step 2: Split of the State

The state \( S(m) \) of the largest \( d_m \) is split into two states, \( S'(m) \) and \( S(M) \), where \( M \) is the current number of the states. Both \( S'(m) \) and \( S(M) \) have the output PDFs of single Gaussian while \( S(m) \) had a PDF of 2-mixture Gaussian. Each PDF of single Gaussian corresponds to each Gaussian of the 2-mixture PDF.

There are parallel splitting called contextual split and sequential one called temporal split in the phoneme SSS. We choose either by estimating maximum likelihood of both contextual split \( P_c \) and temporal split \( P_t \).

--- Contextual Split --- \( S'(m) \) and \( S(M) \) are concatenated in parallel by the contextual split (see Figure 1). Since all paths on \( S(m) \) should be split into two, all training samples should also be split. The samples are split by a contextual factor (such as preceding phonemes or succeeding phonemes) to maximize \( P_c \), which is estimated by the following equation.

\[
P_c = \max_j \sum_max_l \max (p_m(y_{jl}), p_M(y_{jl})),
\]

\( y_{11}y_{21}...y_{T_l} \): \( l \)-th sequence (sample),
\( T \): the length of \( l \)-th sequence,
\( J \): a series of positions of \( l \)-th sequence on \( S(m) \),
\( y_{jl} \): a subsequence of \( y_{11}y_{21}...y_{T_l} \) on \( S(m) \),
\( p_m(y_{jl}) \): total likelihood for \( y_{jl} \) on \( S'(m) \),
\( p_M(y_{jl}) \): total likelihood for \( y_{jl} \) on \( S(M) \).

After maximum \( J \) is determined, corresponding residues \( e_{jl} \) are distributed to the states \( S'(m) \) and \( S(M) \) by the following equation.

\[
\begin{align*}
  e_{jl} &\in E_{m_j} (p_m(y_{jl}) \leq p_M(y_{jl})), \\
  e_{jl} &\in E_{M_j} (p_m(y_{jl}) > p_M(y_{jl})),
\end{align*}
\]

\( E_{m_j} \): a series of residues on \( S'(m) \),
\( E_{M_j} \): a series of residues on \( S(M) \).

--- Temporal Split --- \( S'(m) \) and \( S(M) \) are concatenated in series by the temporal split (see Figure 1). There are two models: \( S'(m) \)-\( S(M) \) and \( S(M) \)-\( S'(m) \). Thus, \( P_t \) is estimated by the following equation.

\[
P_t = \max (p_{mM}(Y), p_{Mm}(Y)),
\]

\( Y \): all samples on \( S(m) \),
\( p_{mM}(Y) \): total likelihood of the model \( S'(m) \)-\( S(M) \),
\( p_{Mm}(Y) \): total likelihood of the model \( S(M) \)-\( S'(m) \).

Step 3: Retraining of the Model

The model which contains two states with single Gaussian PDF, \( S'(m) \) and \( S(M) \), should be retrained by all samples, so that all states have PDFs of 2-mixture Gaussian again.
Step 4: Change of Distribution

When the number of states is increased to a prescribed number, the model is retrained as all states have single Gaussian PDFs.

The computational cost of the original phoneme SSS algorithm is practical. An upper limit is set to the number of the states, which increases by one when the split occurs at every cycle. Thus the number of iteration of this algorithm is limited.

It needs moderate computation at every step of the cycle: estimations of the size of the distribution at all states (Step 1), calculations of $P_c$ and $P_t$ (Step 2), and a training of the HMM (Step 3). Step 1 and Step 2 are rather negligible, and Step 3 has several practical algorithms such as the Baum-Welch algorithm we chose. The cost of the continuous protein SSS will be practical, because the protein SSS is almost the same algorithm as the original one.

2.2 Quantification of Amino Acids

Original SSS algorithm is based on the continuous distribution of phonemes. Takami and Sagayama uses 34 dimensional $VQ$ codes for every (5 msec) frame$[8]$. We apply it to proteins by mapping amino acids onto continuous space. We use Dayhoff PAM-250 matrix as distance matrix and make 19 dimensional space from 20 amino acids at first. Then we estimate two principal axes and map the 19D space onto that plane (see Figure 2). To reduce computational cost, we are implementing continuous SSS algorithm for proteins by this 2D quantification experimentally.

![Amino Acids in PAM-250 onto principal 2 axes](image)

Figure 2: Amino Acids in PAM-250 onto principal 2 axes

2.3 Superfamily and SSS

We should compare between superfamily and SSS in the following two points. One is about the direction of classification. Superfamily is *agglomerative* classification while SSS is *divisive* one. In SSS, we choose the best node to split by estimating $d_i$ for each node $i$
(see Step 1). It indicates the width of whole distribution and the difference of averages of each distribution of 2-mixture Gaussians. If parallel splitting is chosen, it suggests further classification at that point.

The other is the common criterion between classification and abstraction. SSS and HMM commonly use the maximum likelihood criterion. Superfamily uses scores of DP-matching, which is not always the same criterion as multiple alignment.

3 Future Work

We are improving SSS algorithm for proteins. We should evaluate both continuous SSS and discrete SSS. We should know what kind of quantification is the most appropriate for amino acids. We are experimentally using 2D quantification, but we may have to use 19D to keep distance ratios in the PAM, or we may have to introduce other factors such as hydrophobicity or inside-outside index.

Discrete Protein SSS When we apply SSS algorithm to the protein classification, we can think of a discrete protein SSS using multiple alignment as another idea. It is essential that proteins can be split into two groups while it is not so essential that proteins can be regarded as mixture of any parametric PDF such as Gaussian. At Step 0, it is rather reasonable to split proteins according to the multiple alignment scores, though the search for optimal partition of size N needs at least $2^{N-1} - 1$ trials of two group alignment. It is invaluable to discover good estimation to reduce computation.

At Step 1, we can use entropy for the decision criterion instead of $d_i$. It can be estimated by $D_m$ in the following equation:

$$D_m = \sum_r p_r \log p_r,$$

where $p_r$ is the frequency of the residue $r$.

At Step 2, the parallel split corresponds to the distribution of protein sequences into two groups. We should somehow reduce the number of trials of such splitting, because there are $2^{N-1} - 1$ ways where $N$ is the number of proteins. It is the same problem as the one at Step 0.

The sequential split corresponds to the vertical partition of the alignment. There are not so many more than $L$ ways of partitions where $L$ is the length of the alignment. In other words, there are not more than $2L'$ ways where $L'$ is the length of the longest sequence. It is the rather reasonable number of trials than the parallel split, though every trial needs one alignment procedures for each part of sequences.

The discrete protein SSS algorithm is impractical unless we have any idea to reduce computation at Step 0 and Step 2.

Continuous Protein SSS: other scalings for the amino acids We can use various indexes such as the electric charge or the hydrophobicity instead of the space made from PAM-250. Through an investigation of the resultant HMM, we can recognize what kind of indexes are influential for every specific characteristics, such as alpha-helix domain of secondary structures or DNA binding sites of functional domains.
4 Concluding Remarks

The SSS algorithm enables to obtain an appropriate Hidden Markov Network automatically, and the Hidden Markov Model simultaneously. Its effectiveness is confirmed in phonemes [8]. The proteins will be classified by SSS algorithm without previous knowledge of the proteins. We are applying it to the protein classification by regarding amino acids as a continuous variant and implementing the continuous protein SSS algorithm.

References


