Representing Inter-residue Dependencies in Protein Sequences with Probabilistic Networks

Hiroshi Mamitsuka
mami@sbl.cl.nec.co.jp

C&C Research Labs., NEC Corporation
1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki, Kanagawa 216, Japan

Abstract

We propose a new method for representing a local region of a protein sequence as a probabilistic network. The method produces, from a large number of examples of a local region, a network which describes dependency relationships that exist among amino acid residues in the region. The network is constructed using the greedy-search algorithm based on the minimum description length (MDL) principle. In our experiments, we construct two probabilistic networks of two α-helix regions in globin family protein. Experimental results show that our method provides a visual aid to understanding inter-residue dependencies of those regions with probabilistic networks, and the networks capture several important features which are peculiar to those regions.
1 Introduction

Interactions among amino acid residues in a protein sequence may prove to be a crucial factor for determining its three-dimensional structure or defining its own function. In representing inter-residue relationships in protein sequences, the central issue is choosing a good mathematical model.

We here focus on inter-residue dependency relationships in protein sequences, and propose a new method for constructing a probabilistic network which can provide visual insights into probabilistic dependency relationships hidden among residues.

We address this issue for the following two reasons:

1) Visual representation

We believe that a clear representation of inter-residue dependencies of a protein sequence allow us to precisely understand functional or structural mechanism of that protein.

2) Methods for representing inter-residue dependency relationships

Although a number of approaches have been proposed to date for seeking a suitable model for representing interactions among residues, no method has ever provided visual insights into dependency relationships among residue positions.

Actually, for instance, neural network learning methods ([11], etc.) have not presented any dependency relationships among residues. Similarly, a number of approaches which can deal with several residues as a joint event, such as a form of mutual information ([3],[5]), also cannot have treated inter-residue dependencies.

It is in response to these considerations, our method for constructing probabilistic networks which can represent inter-residue dependency relationships has following two unique characteristics:

1) Probabilistic network

Our method constructs probabilistic networks in which each node corresponds to a residue position. Probabilistic dependency relationships among nodes are visually presented by directed arcs in the probabilistic networks, and the strengths of these relationships are quantified by conditional probabilities. The dependencies provided in the networks are typically recognized as the causal relationships among the nodes connected by the arcs ([8]). Thus, from these probabilities, we can analyze further inter-residue dependencies which are found by our method.

2) Efficient search

Our method for constructing a probabilistic network is characterized by the greedy-search algorithm and the minimum description length (MDL) principle ([12],[13]). Using the probabilities which are directly calculated from given examples, we can efficiently determine the residues on which each residue of a local region is dependent, based on the greedy-search. This strategy requires greatly little time to construct a probabilistic network structure.

The remainder of this paper is organized as follows. In Section 2, we present a probabilistic network and propose an efficient algorithm for constructing the networks. In Section 3, we report the results of experiments in which we apply our method to actual regions in protein sequences to construct probabilistic networks. The experimental results indicate that the constructed networks express important features of the corresponding regions. Section 4 concludes this paper with a brief summary and a discussion of open problems.
2 Methods

2.1 Probabilistic Network

In order to model dependency relationships that exist among amino acid residues, we focus on using a "Bayesian belief network (or causal probabilistic network)" ([8],[7]). Hereafter we simply refer it to as a probabilistic network.

In this subsection, we describe typical properties of probabilistic networks.

A probabilistic network is a directed acyclic graph (often simply referred to as a DAG) in which nodes represent domain variables and arcs which connect between nodes represent probabilistic dependencies of variables of corresponding nodes.

Figure 1 shows an example of a probabilistic network which consists of three variables: $x_1$, $x_2$ and $x_3$.

![Diagram of a probabilistic network with nodes $x_1$, $x_2$, and $x_3$ and arcs connecting them]

Figure 1: An example of a probabilistic network

For instance, in this figure, the arc from $x_1$ to $x_2$ indicates a probabilistic dependency between these two variables, and the absence of the arc from $x_2$ to $x_3$ implies that there does not exist a direct dependency between them. Furthermore, the strength of a probabilistic dependency between $x_1$ and $x_2$ is quantified in terms of a conditional probability as follows:

$$P(x_2|x_1)$$

(1)

When an arc is connected between two nodes, such as an arc from $x_1$ to $x_2$ in Figure 1, a node which is a predecessor of the arc is called a parent node and the node which is a descendant of the arc is called a child node. We use $x_i$ to denote parent nodes of $x_i$. Then a conditional probability of $x_i$ is typically given as follows:

$$P(x_i|x_i)$$

(2)

A joint probability of all the nodes in a network can be calculated as a product of the conditional probability at each node. For example, a joint probability of three nodes corresponding to Figure 1 is calculated as follows:

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1)$$

(3)

A joint probability of $n$ nodes in a probabilistic network is generally given by the following formulas:

$$P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i|x_i)$$

(4)

In the probabilistic network which represents a protein sequence, each node corresponds to an amino acid residue position, and variable values are 20 types of amino acids or several groups of amino acids.
2.2 Heuristic Method

In trying to construct a probabilistic network which represents inter-residue dependency relationships in a particular region, most crucial problem lies in how we determine parent nodes for each node of the network.

In addition, most of the conventional methods for constructing probabilistic networks have proposed time-consuming approaches ([8],[7]), since they tried to find an optimal network from all possible network structures, and required quite a bit of time to searching for the optimal one.

To solve these problems, we use the minimum description length (MDL) principle ([12],[13]) and the greedy-search heuristics.

The MDL principle insists that an optimal network is the one which minimizes

\[
\text{(description length for examples relative to a given network)} + \text{(description length for the network itself)}.
\]

We further assume that the description length for a probabilistic network can be written as the sum of the description length for probabilistic dependencies at each node. Under the constraint, minimizing the description length for a network yields minimizing one for each node. Thus, instead of checking the description length for all types of networks, all we need to do is to determine, for each node, the nodes to obtain the minimum description length when they are parent nodes.

In addition, we search the parents nodes for each node on the basis of the greedy heuristics. This framework is similar to the one which is proposed by Cooper and Herskovits ([11]). They tried to determine parent nodes for every node using the greedy-search algorithm based on their own criteria. However, before determining parent nodes for each node, their method needs to have an ordering on the nodes in order to avoid cyclic dependency among nodes, and the node ordering greatly restricts the network structure to several limited ones.

In contrast with their method, we can construct a directed acyclic graph without any constraint, such as the node ordering, since our algorithm can sequentially determine the parent node on which each node is dependent, without producing directed cycles.

Our method consists of three phases: example generation, parameter estimation, and learning. The following procedure expresses our algorithm.

1. Example generation

We use both positive and negative examples of a protein local region (hereafter referred to as a target region) within which we wish to represent inter-residue dependencies by a probabilistic network.

Positive examples consist of corresponding regions in those proteins which belong to the same family or have the same name as the one comprises the target region, when each of them is aligned.

Negative examples consist of the local sequences which are not in positive examples and highly homologous to the target region. See [6] for the details of the method to generate examples.

2. Parameter estimation

At each residue position of a target region, we categorize 20 amino acids into roughly 10 to 15 groups based on the MDL principle. Also see [6] for the way to classify 20 amino acids using the MDL principle.

Categorizing amino acids into more or less 10 groups makes us allow to calculate joint probabilities and conditional probabilities of residue positions, even when only a small
number of examples are obtained at the example generation phase. This is attributed to the fact that the MDL principle successfully optimizes the number of groups depending on the number of given examples. That is, the smaller the number of given examples, the smaller the number of groups can be categorized by the MDL principle.

We denote the number of groups of amino acids at the i-th position of a target region to be \( m_i \). We further use a Laplace estimator ([14]) to calculate joint probabilities and conditional probabilities from obtained examples.

3. Learning

Here, in order to determine parent nodes of each node in a target region, we focus on the t-th node and describe the way which we here use to calculate the description length at the node.

Let the number of parent nodes at the t-th node be \( k \). Let the number of examples in which an amino acid of \( x_t \) is \( i \) and amino acids of \( \pi_t \) are \( j \) be \( N(x_t^i, \pi_t^j) \), and let the number of positive examples in which an amino acid of \( x_t \) is \( i \) and amino acids of \( \pi_t \) are \( j \) be \( N^+(x_t^i, \pi_t^j) \). Let the conditional probability when an amino acid of \( x_t \) is \( i \) and amino acids of \( \pi_t \) are \( j \) be \( p(x_t^i|\pi_t^j) \), and let the h-th node of the parent nodes be \( t_h \).

The description length for given examples is calculated as follows:

\[
- \sum_{j_1=0}^{m_t} \ldots \sum_{j_k=0}^{m_t} \sum_{i=0}^{m_t} \log \{ p(x_t^i|\pi_t^{j_1\ldots j_k}) N^+(x_t^i, \pi_t^{j_1\ldots j_k}) (1 - p(x_t^i|\pi_t^{j_1\ldots j_k})) N(x_t^i, \pi_t^{j_1\ldots j_k}) - N^+(x_t^i, \pi_t^{j_1\ldots j_k}) \} (5)
\]

Note that this is not exact expression of the description length for given positive and negative examples. Precise description length for given examples is given as more complicated calculas than (5). For simplicity, we here use (5) for calculating the description length for given examples.

The description length for the rule itself is calculated as follows:

\[
\sum_{j_1=0}^{m_t} \ldots \sum_{j_k=0}^{m_t} \sum_{i=0}^{m_t} \log \frac{N(x_t^i, \pi_t^{j_1\ldots j_k})}{2} (6)
\]

Therefore, the total description length at the t-th node is calculated as follows:

\[
- \sum_{j_1=0}^{m_t} \ldots \sum_{j_k=0}^{m_t} \sum_{i=0}^{m_t} \log \{ p(x_t^i|\pi_t^{j_1\ldots j_k}) N^+(x_t^i, \pi_t^{j_1\ldots j_k}) (1 - p(x_t^i|\pi_t^{j_1\ldots j_k})) N(x_t^i, \pi_t^{j_1\ldots j_k}) - N^+(x_t^i, \pi_t^{j_1\ldots j_k}) \} + \sum_{j_1=0}^{m_t} \ldots \sum_{j_k=0}^{m_t} \sum_{i=0}^{m_t} \log \frac{N(x_t^i, \pi_t^{j_1\ldots j_k})}{2} (7)
\]

According to the MDL principle, optimal parent nodes for each node is estimated as those which minimize (7). We use (7) for calculating the description length of a network, whom the network is being constructed based on the greedy heuristics.
Below, we present the procedure `net_learn()` which determines the network structure from given examples.

Here we newly define two types of nodes, i.e., a descendant node and an ancestor node. A descendant node is recursively defined. A child node is a descendant node, and a child node of a descendant node is a descendant node. An ancestor node is also recursively defined. A parent node is an ancestor node, and a parent node of an ancestor node is an ancestor node.

In this procedure of the greedy-search, we use a simple idea that any directed cycle is not generated under the condition that any descendant node of a node is not added to its parent nodes.

\[
\begin{align*}
\text{par}(i) & \text{ := parent nodes of the } i\text{-th node} \\
\text{des}(i) & \text{ := descendent nodes of the } i\text{-th node} \\
\text{new}(i) & \text{ := a node which minimizes the description length when it is added to parent nodes of the } i\text{-th node, and if there does not exist such a node, } 0 \text{ is given to } \text{new}(t) \\
\text{dif}(i) & \text{ := the difference of the description length when the } \text{new}(t) \text{ is not } 0 \text{ and is added to parent nodes of the } i\text{-th node} \\
N & \text{ := the number residues in a target region} \\
dl(i) & \text{ := the description length at the } i\text{-th node}
\end{align*}
\]

for \( i := 1 \) to \( N \)
Calculate \( dl(i), \text{new}(i), \text{dif}(i) \) when the \( i\text{-th node has no parent node} \)

\[
\text{net_learn}(N, \text{par}(), \text{des}(), \text{new}(), \text{dif}(), dl())
\]

\[
\begin{align*}
n & := 0 \\
\max & := 0.0 \\
i & := 1 \\
\text{while } i \text{ is not more than } N \\
\quad \text{if } \text{new}(i) \text{ is } 0 \text{ then} \\
\quad \quad n & := n + 1 \\
\quad \text{else} \\
\quad \quad \text{if } \text{dif}(i) \text{ is larger than } \max \text{ then} \\
\quad \quad \quad \max & := \text{dif}(i) \\
\quad \quad \quad \text{cn} & := i \\
\quad \quad \quad \text{pn} & := \text{new}(i) \\
\quad \quad i & := i + 1 \\
\quad \text{if } n \text{ is not } N \text{ then} \\
\quad \quad \text{add } \text{pn}\text{-th node to } \text{par}(\text{cn}) \\
\quad \quad \quad \text{dl}(\text{cn}) & := \text{dl}(\text{cn}) - \text{dif}(\text{cn}) \\
\quad \quad \text{calculate } \text{des}() \text{ of ancestor nodes of the } \text{cn}\text{-th node} \\
\quad \quad \text{calculate } \text{dif}() \text{ and } \text{new}() \text{ of ancestor nodes of the } \text{cn}\text{-th node} \\
\quad \quad \text{net_learn}(N, \text{par}(), \text{des}(), \text{new}(), \text{dif}(), dl())
\end{align*}
\]

output \( \text{par}() \)

This procedure is executed until the parent nodes \( \text{par}() \) are outputted.
3 Experimental Results

We use two $\alpha$-helix regions contained in globin family proteins as target regions. Two regions correspond to residues 36 to 42 (hereafter referred to as region A) and residues 80 to 88 (hereafter referred to as region B) in human hemoglobin $\alpha$ subunit.

From the SWISS-PROT database, we select 514 sequences, which are hemoglobin $\alpha$ and $\beta$ subunits and myoglobin, to align each of them. The regions in those sequences, which correspond to two regions, are considered to be positive examples respectively.

We select, from the Brookhaven Protein Data Bank (PDB) database, the same number of negative examples as positive ones for two regions respectively, under the following conditions: 1) they are 30-40% homologous in its primary structure to and have the same number of residues as two regions, and 2) they are not in a non-$\alpha$-helix region.

For these two regions, we construct probabilistic networks based on the method noted in the earlier section. Constructed networks are shown in Figure 2. The computation time for constructing these networks is 1.0 second for region A and 1.5 second for region B on NEC EWS4800/260. Both figures show that the number of parent nodes of all the nodes in these regions is at most 2. Our method implies that, in local regions, any residue can be dependent on only 0 to 2 residues.

![Figure 2: Constructed probabilistic networks](image)

Axial projection forms of two $\alpha$-helix regions are also shown in Figure 3.

Below, we analyze these networks representing inter-residue dependencies of region A and B on the basis of the observation on dynamic three-dimensional structures.

**Region A**

Hemoglobin takes two conformations, i.e. deoxy-hemoglobin (hereafter referred to as the T-state) and oxy-hemoglobin (hereafter referred to as the R-state), and ionic bonds between $\alpha$ subunit and $\beta$ subunit drastically vary according to the conformational change from the T-state to the R-state (hereafter referred to as the $T-R$ conformational change) ([9],[10]). In particular, residue 38 and residue 41 which face on residues 93 to 99 in $\beta$ subunit are fundamental to
the T-R conformational change ([2]). These two residues shows the following variations in the T-R conformational change. In the T-state, residue 41 closes to residue 97 or 99 in β subunit, though, in the R-state, instead of residue 41, residue 38 closes to residue 94 or 97 in β subunit.

Residue 37 and 42 which are located around residue 38 and 41 in three-dimensional structure cooperatively work upon ionic bonds between α subunit and β subunit.

Figure 2 (a) and Figure 3 (a) show that there exist successive direct dependencies from residue 38 to residue 37 (38 → 41 → 42 → 39 → 37) which are included in the residues relating to ionic bonds between α subunit and β subunit. These dependencies indicate that residue 37, 39, and 42 (hereafter referred to as group A1) depend on residue 38 and 41 (hereafter referred to as group A2).

This result implies that, in region A, group A2 is the dominant factor for the ionic bonds between α subunit and β subunit. Furthermore, the network allows us to further observe inter-residue dependency relationships in region A. For example, residue 38 comprises direct dependency relationships with residues 36 and 41, though residue 37 and 39 are neighbors of residue 38 in a primary structure.

Region B

Region B corresponds to one of two regions which constitute the "heme pocket" which is an active site of hemoglobin for accepting an iron porphyrin, and this region is one of the most important portions for initiating the T-R conformational change ([4],[9],[10]). In the T-state, there exists a covalent bond between residue 87 and the iron atom. Residue 83, 84 and 86 which are typically occupied by hydrophobic amino acids and, in actual three-dimensional structure, are located near residue 87 extend to the porphyrin ring which surrounds the iron atom, and these three residues and residue 87 control the T-R conformational change.

Figure 2 (b) shows that all the residues except residue 87 are the descendent residues of residue 87, and those residues are dependent on residue 87. According to these dependencies, residue 87 is the crucial residue among the residues contained in region B.

This relationships may explain further details of the interactions between region B and the
heme pocket. That is, residue 87 which can link to the iron atom is the dominant factor for three residues (residue 86, 84 and 83) which can access to the porphyrin ring. This result implies that the conformational interaction of porphyrin and the surface of region B would determine the bond between residue 87 and the iron atom. Moreover, the network also presents inter-residue dependencies in region B.

4 Discussion and Conclusions

We have shown in our experiments that the probabilistic networks constructed by our method present visual inter-residue dependency relationships. We have also shown from observation on protein three-dimensional structures that the constructed networks capture significant dependencies existing among amino acid residues within several important functions of the regions used as training examples. In particular, successive dependencies found in both region A and B indicate that our method can be used to search for unknown dependency relationships hidden in protein sequences.

Furthermore, our method determines parent nodes for each node on the basis of the greedy-search algorithm and the MDL principle, and it requires significantly shorter learning time than the other learning methods in which iterative algorithms are used, such as a back-propagation of neural network learning. Parallel computation of the description length at each node will greatly shorten the learning time of our method.

Each conditional probability in probabilistic networks used here can be regarded as a probabilistic rule of each residue position in given target regions. In the α-helix region prediction method presented in [6], all residues in an α-helix region are assumed to be probabilistically independent. However, each conditional probability calculated here comprise inter-residue dependencies in an α-helix region. The prediction accuracy in [6] may be improved by applying the probability to it as a rule for each residue position in α-helix regions. Applying the probabilistic networks presented here to predicting protein local regions will be one of important future works.

Second important future work consists in using a precise expression for calculating the description length for given examples. In this paper, we calculate the description length with a simple form which is not theoretically exact approximation of the description length for given positive and negative examples. Using a more precise expression of the description length instead of the one used in this paper will allow us to obtain different probabilistic networks which may provide more crucial information hidden in protein sequences.

Acknowledgements

The author wishes to thank Dr. Kenji Yamanishi of the NEC Research Institute for helpful comments on the calculation of the description length; Mr. Jun-ichi Takeuchi for introduction to the area of probabilistic networks; Dr. Naoki Abe for suggestion to employ global greedy search in the learning algorithm; Dr. Kazuhiro Iida of Fundamental Research Laboratories, NEC Corporation for helpful advice.

Finally, the author wishes to express his sincere gratitude of Mr. Katsuhiro Nakamura and Mr. Akihiko Konagaya for their support and helpful advice.

References


