VOL. 3 NO. 2 September 2014 - ISSN 2233 – 1859

UOIuBIH
ORSinBIH
Operations Research Society in Bosnia and Herzegovina

Southeast Europe Journal of Soft Computing

Available online: www.scjournal.com.ba

IUS Soft Computing Research Group

Predicting the Secondary Structure of Proteins Using Artificial Neural Networks

Betul Akcesme and Faruk B. Akcesme
International University of Sarajevo, Faculty of Engineering and Natural Sciences, Hrasnicka Cesta 15, Ilidža
71210 Sarajevo, Bosnia and Herzegovina
betul.cicek@yahoo.com; fakcesme@ius.edu.ba

Article Info

Article history:
Article received Sep. 2014
Article recivied in revised form Nov. 2014

Keywords:
Secondary structure; Conformation of proteins; Statistical methods

Abstract

A method for protein secondary structure prediction based on the use of artificial neural networks (ANN) is presented. Amino acids, and their secondary structures obtained from National Center for Biotechnology Information (NCBI) and the online tool given in Chou-Fasman website of seven proteins are concatenated to create a sequence of 15536 residues. A neural network with only an input and an output layer is used, and back-propagation technique is adopted to tune the synaptic weights. Data is divided into two sets for training, and testing. The average success rate of the method on a testing set of proteins was 90.64% in training and 89.13% in testing on three types of secondary structure α-helix, β-sheet, and coil, with correct identification coefficients of $C_\alpha = 0.92$, $C_\beta = 0.81$, and $C_{coil} = 0.82$, and $T_{turn} = 0.81$. These quality indices are all compatible with those of previous methods. From computational experiments on real and artificial structures that no method based solely on local information in the protein sequence is likely to produce significantly better results for proteins.

1. INTRODUCTION

Our knowledge about protein structure comes mostly from the X-ray diffraction patterns of crystallized proteins, NMR spectroscopy and electron microscopy. X-ray crystallography is essentially very accurate, but many steps are uncertain since not all proteins can easily be crystallized. Obtaining high-quality protein sample is difficult and generally proteins are sensitive to temperature and pH. All these techniques are very time consuming and costly.

Recent developments in genetic engineering have vastly increased the number of known protein sequences. In addition, it is now possible to selectively alter protein sequences by site-directed mutagenesis. But to take full advantage of these techniques would be helpful if one could predict the structure of a protein from its primary sequence of amino acids. The general problem of predicting the tertiary structure of folded proteins is unsolved.

Information about the secondary structure of a protein can be helpful in determining its structural properties. The best way to predict the structure of a new protein is to find a homologous protein whose structure has been determined. Structure of new protein can be found with many available online tools that use protein database. Even if only limited regions of conserved sequences can be found, then template matching methods are applicable (Taylor, 1986). If no homologous protein with a known structure is found, existing methods for predicting secondary structures can be used but are not always reliable. Three of the most commonly used methods are those of Robson (Robson &

Pain, 1971; Garnier et al., 1978), of Chou & Fasman (1978), and Lim (1974). These methods primarily exploit, in different ways, the correlations between amino acids and the local secondary structure. By local, we mean an influence on the secondary structure of an amino acid by others that are no more than about ten residues away. These methods were based on the protein structures available in the 1970s. The average success rate of these methods on more recently determined structures is 50 to 53% on three types of secondary structure (α-helix, β-sheet, and coil: Nishikawa, 1983; Kabsch & Sander, 1983a).

In this paper, we have employed a method for discovering regular patterns in data that is based on neural network models. The brain has highly developed pattern matching abilities and neural network models are designed to mimic them.

The goal of the method introduced here is to use the available information in the database of known protein structures to help predict the secondary structure of proteins for which no homologous structures are available in any database. The known structures implicitly contain information about the bio-physical properties of amino acids and their interactions. This approach is not meant to be an alternative to other methods that have been developed to study protein folding that take biophysical properties explicitly into account, such as the methods of free energy minimization (Scheraga, 1985) and integration of the dynamical equations of motion (Karplus, 1985; Levitt, 1983). Rather, secondary structures obtained using ANN provides additional constraints to reduce the search space for these other methods. For example, a good prediction for the secondary structure could be used as the initial conditions for energy minimization, or as the first step in other predictive techniques (Webster et al., 1987).

2. METHODS

(a) Database

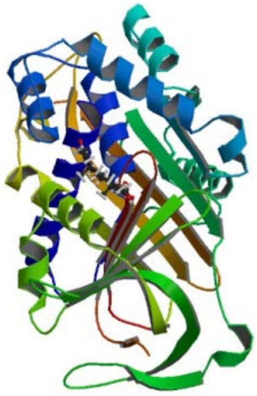
Primary structures of seven proteins are obtained from the NCBI. Predicted secondary structures of these proteins are obtained from the online tool given in Chou-Fasman website[1]. Amino acid residues and their secondary structure assignments are concatenated to create a data sequence of 15536 amino acids. Table 1 contains a listing of the seven proteins that were used in this study.

	Protein	Residue
1.	Serum Albumin	1500
2.	Spondin1, extracellular matrix protein	807
3.	Collagen type IV	1779
4.	Cystic fibrosis transmembrane c. regulator	1483
5.	Pod1,isoform G	1266
6.	Polyprotein [Hepatitis C virus genotype 3]	3021
7.	Dystrophin	3678
	Total	5536

Table 1 The seven proteins that were used in this study

Data is divided into three equal sets for training, validation and testing. Results are highly sensitive to protein homologies in the testing and training sets. Special care was taken to balance the overall frequencies of α-helix, β-sheet in the training and testing sets, as shown in Tables 2.

Figure 1 α-helices, β-sheets, and coils on the same picture.



(PDB code for the proteins: 4C49)

	Number of residues		
Estimator	C-F	GOR	ANN
α-helix	4919	5529	4221
β-sheet	3923	2697	1149
Coil	4992	2895	8164
Turn		2413	

Table 2 Balance the overall frequencies of α-helices, β-sheets in the training and testing sets

(b) Symbols for Amino Acids

Proteins are chains in the three dimensional space built

#	Amino acid	Chemical	alphabet
1	Alanine	Ala	A
2	Arginine	Arg	R
3	Asparagine	Asn	N
4	Aspartic acid	Asp	D
5	Cysteine	Cys	C
6	Glutamine	Gln	Q
7	Glutamic acid	Glu	E
8	Glycine	Gly	G
9	Histidine	His	H
10	Isoleucine	Ile	I
11	Leucine	Leu	L
12	Lysine	Lys	K
13	Methionine	Met	M
14	Phenylalanine	Phe	F
15	Proline	Pro	P
16	Serine	Ser	S
17	Threonine	Thr	T
18	Tryptophan	Trp	W
19	Tyrosine	Tyr	Y
20	Valine	Val	V

Table 3 Names and symbols of 20 amino acids

[1] http://cib.cf.ocha.ac.jp/bitool/MIX/

from smaller chemical molecules called amino acids. There are 20 different amino acids. Each of them is denoted by a different letter in the Latin alphabet as shown in Table 3.

Based on the protein chain it is easy to create its relevant sequence of amino acids replacing an amino acid in chain by its code in Latin alphabet. As a result a word on the amino acids’ alphabet is received. This word can be called a protein primary structure on the condition that letters in this word are in the same order as amino acids in the protein chain are.

A secondary structure of a protein is a subsequence of amino acids coming from the relevant protein. These subchains form in the three dimensional space regular structures which are the same in shape for different proteins. In the analysis, a similar representation for the secondary structures as for the primary ones has been used. A secondary structure is represented by a word on the relevant alphabet of secondary structures. Each kind of a secondary structure has its own unique letter α-helix, H; β-sheet E, and coil C. An alphabet of secondary structures consisting of three different secondary structures has been considered in the analysis.

(c) Coding the Data

In this paper, data corresponding to an amino acid consists of six right, and six left neighboring amino acids of this amino acid in the primary structure of the protein as in Table 3. In the second row, secondary structure conformations of these neighboring amino acids are given.

A	E	E	K	E	A	V	L	G	L	W	G	K
H	H	H	H	H	E	E	E	E	C	C	C	E
1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	0	0	1	1	1	0

Table 4 Six right, and six left neighboring amino acids of the amino acid V

Secondary structure letters H, E, and C are coded as in the table below;

H	E	C
1	0	0
0	1	0
0	0	1

Table 5 Codes for secondary structure letters H, E, and C.

The data corresponding to an amino acid is coded by a 20×13 matrix as follows

	A	E	E	K	E	A	V	L	G	L	W	G	K
A	1	0	0	0	0	1	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	1
D	0	0	0	0	0	0	0	0	0	0	1	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	1	1	0	1	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	1	0	0	1	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	1	0	1	0	0	0
K	0	0	0	1	0	0	0	0	0	0	0	0	1
M	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	1	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	1	0	0	0	0	0	0

Table 6 The data corresponding to the central amino acid V

(d) ANN Architecture

Nervous systems existing in biological organism have been the subject of studies for mathematicians who tried to develop some models describing such systems and all their complexities for years. Artificial Neural Networks emerged as generalizations of these concepts with mathematical model of artificial neuron due to McCuloch and Pitts described in 1943 (McCuloch and Pitts 1943) definition of unsupervised learning rule by Hebb in 1949 (Heb 1949), and the first ever implementation of Rosenblatt’s perceptron in 1958 (Rosenblatt 1958). The efficiency and applicability of artificial neural networks to computational tasks have been questioned many times, especially at the very beginning of their history the book "Perceptrons" by Minsky and Papert (Minsky and Papert 1969), published in 1969, caused dissipation of initial interest and enthusiasm in applications of neural networks.

It was not until 1970s and 80s, when the back propagation algorithm for supervised learning was documented that artificial neural networks regained their status and proved beyond doubt to be sufficiently good approach to many problems. Artificial Neural Network can be looked upon as a parallel computing system comprised of some number of rather simple processing units (neurons) and their interconnections. They follow inherent organizational principles such as the ability to learn and adapt, generalization, distributed knowledge representation, and fault tolerance. Neural network specification comprises definitions of the set of neurons (not only their number but also their organization), activation states for all neurons expressed by their activation functions and offsets specifying when they fire, connections between neurons which by their weights determine the effect the output signal of a neuron has on other neurons it is connected with, and a method for gathering information by the network that is its learning (or training) rule.

From architecture point of view neural networks can be divided into two categories: feed-forward and recurrent networks. In feed-forward networks the flow of data is strictly from input to output cells that can be grouped into layers but no feedback interconnections can exist. On the other hand, recurrent networks contain feedback loops and their dynamical properties are very important.

The most popularly used type of neural networks employed in pattern classification tasks is the feed forward network which is constructed from layers and possesses unidirectional weighted connections between neurons. The common examples of this category are Multilayer Perceptron or Radial Basis Function networks, and committee machines.

Multilayer perceptron type is more closely defined by establishing the number of neurons from which it is built, and this process can be divided into three parts, the two of which, finding the number of input and output units, are quite simple, whereas the third, specification of the number of hidden neurons can become crucial to accuracy of obtained classification results.

The number of input and output neurons can be actually seen as external specification of the network and these parameters are rather found in a task specification. For classification purposes as many distinct features are defined for objects which are analyzed that many input nodes are required. The only way to better adapt the network to the problem is in consideration of chosen data types for each of selected features. For example instead of using the absolute value of some feature for each sample it can be more advantageous to calculate its change as this relative value should be smaller than the whole range of possible values and thus variations could be more easily picked up by artificial neural network. The number of network outputs typically reflects the number of classification classes.

The third factor in specification of the multilayer perceptron is the number of hidden neurons and layers and it is essential to classification ability and accuracy. With no hidden layer the network is able to properly solve only linearly separable problems with the output neuron dividing the input space by a hyperplane. Since not many problems to be solved are within this category, usually some hidden layer is necessary.

With a single hidden layer the network can classify objects in the input space that are sometimes and not quite formally referred to as simplexes, single convex objects that can be created by partitioning out from the space by some number of hyperplanes, whereas with two hidden layers the network can classify any objects since they can always be represented as a sum or difference of some such simplexes classified by the second hidden layer.

Apart from the number of layers there is another issue of the number of neurons in these layers. When the number of neurons is unnecessarily high the network easily learns but poorly generalizes on new data. This situation reminds auto-associative property: too many neurons keep too much information about training set rather "remembering" than "learning" its characteristics. This is not enough to ensure good generalization that is needed.

On the other hand, when there are too few hidden neurons the network may never learn the relationships amongst the input data. Since there is no precise indicator how many neurons should be used in the construction of a network, it is a common practice to build a network with some initial number of units and when it learns poorly this number is either increased or decreased as required. Obtained solutions are usually task-dependant.

Activation Functions

Activation or transfer function of a neuron is a rule that defines how it reacts to data received through its inputs that all have certain weights.

Among the most frequently used activation functions are linear or semi-linear function, a hard limiting threshold function or a smoothly limiting threshold such as a sigmoid or a hyperbolic tangent. Due to their inherent properties, whether they are linear, continuous or differentiable, different activation functions perform with different efficiency in task-specific solutions.

For classification tasks with more than two classes logistic activation function and its derivative is better:

$$\boldsymbol{\emptyset(z) = \frac{1}{(1+e^{-z})}};$$

$$\emptyset'(z) = \emptyset(1-\emptyset). \qquad (1)$$

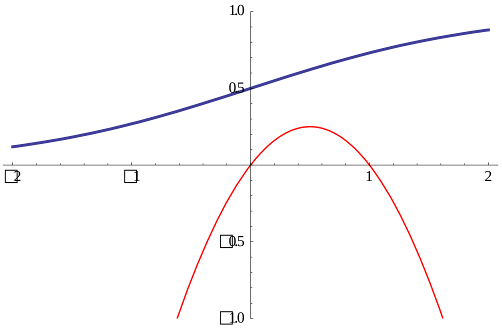


Figure 2 Logistic activation function and its derivative

Learning Rules

In order to produce the desired set of output states whenever a set of inputs is presented to a neural network it has to be configured by setting the strengths of the interconnections and this step corresponds to the network learning procedure. Learning rules are roughly divided into three categories of supervised, unsupervised and reinforce-ement learning methods.

The term supervised indicates an external teacher who provides information about the desired answer for each input sample. Thus in case of supervised learning the training data is specified in forms of pairs of input values and expected outputs. By comparing the expected outcomes with the ones actually obtained from the network the error function is calculated and its minimization leads to modification of connection weights in such a way as to obtain the output values closest to expected for each training sample and to the whole training set.

In unsupervised learning no answer is specified as expected of the neural network and it is left somewhat to itself to discover such self-organization which yields the same values at an output neuron for new samples as there are for the nearest sample of the training set.

Reinforcement learning relies on constant interaction between the network and its environment. The network has no indication what is expected of it but it can induce it by discovering which actions bring the highest reward even if this reward is not immediate but delayed. Basing on these rewards it performs such re-organization that is most advantageous in the long run (McCulloch, and Pill's 1943).

The modification of weights associated with network interconnections can be performed either after each of the training samples or after finished iteration of the whole training set.

The important factor in this algorithm is the learning rate η whose value when too high can cause oscillations around the local minima of the error function and when too low results in slow convergence. This locality is considered the drawback of the back propagation method but its universality is the advantage.

Perceptrons

As the base topology of artificial neural network (Tang et. Al. 2007) with the feed-forward simple perceptron with logistic activation function trained by back propagation algorithm is used.

In this research a perceptron with one input layer with 20×13 ports and one output layer with three output neurons is used. Feed forward technique is employed, and artificial neural network is trained by back propagation. The three output neurons communicate and the winner neuron defines the conformation of the amino acid in the center of 13 neighboring amino acids.

Feeding Forward

Given $W_{ij}^{k},\ i = 1, \dots, 20;\ j = 1, \dots, 13, ; k = 1,2,3$ and $W_k^0, k = 1,2,3$, Out(1), Out(2), and Out(3) are computed according to the formulas in (2). After the application of the activation function $\emptyset$**,** the position of the largest, gives the type of the conformation of the central amino acid.

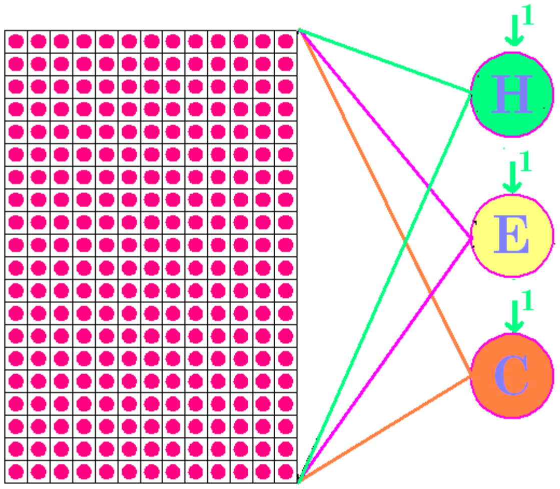


Figure 3 Preceptor with one input layer with 20X13 ports, and one output layer with three output neurons

$$\mathrm{Out}(1) = \sum W_{ij}^{1}\, x_{ij} + W_1^0$$

$$\mathrm{Out}(3) = \sum W_{ij}^{2}\, x_{ij} + W_3^0$$

$$\mathrm{Out}(3) = \sum W_{ij}^{2}\, x_{ij} + W_3^0 \qquad (2)$$

$$\mathrm{Out} = \mathrm{Max}\Big(\emptyset\big(\mathrm{Out}(1), \mathrm{Out}(2), \mathrm{Out}(3)\big)\Big) \qquad (3)$$

$$\mathrm{Conformation} = \mathrm{Position}(\mathrm{Out}) \qquad (4)$$

Back Propagation

When all of *n* data points are exposed to the perceptron and output vector *out* is obtained as a 3×*n* matrix of which a part is of the form;

H	C	H	H	H	E	E	C	E	C	E	C	E
1	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	1	0	1	0	1
0	1	0	0	0	0	0	1	0	1	0	1	0

Assume that for the training data the known conformation is

H	H	H	H	H	E	E	E	E	C	C	C	E
1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	0	0	1	1	1	0

Subtracting these three rows from the previous three rows, in absolute value, we get a part of the *error matrix*:

0	1	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	1	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	1	0	0

The sum of the elements of this matrix after division to the twice the number of residues in this part of the protein can be taken as a measure for the error caused by the synaptic weights $W_{ij}^{k},\ i = 1, \dots ,20;\ \ j = 1, \dots ,13, ; k = 1,2,3$ and $W_k^0, k = 1,2,3.$

$$error = 6/26 \approx 0.230769$$

which is the ratio of the misclassifications. Then this error is back propagated to adjust the synaptic weights.

$$
\begin{aligned}
&\quad Do[\{v1[[jj]] \\
&\quad = Table[Sum[train[[jj,k]].w1[[i,k]],\{k,1,na\}] \\
&\quad + w0[[i]],\{i,1,str\}], \\
&y1[[jj]] = phi\left[v1[[jj]]\right], Mt = Max\left[y1[[jj]]\right], \\
&ss = Position[y1[[jj]],Mt][[1,1]], \\
&y2 = ReplacePart[id0,ss \to 1], \\
&e[[jj]] = tid[[jj]] - y2\},\{jj,1,n1\}], \\
&te = Total\left[Abs[Transpose[e]]\right], \\
&error = 1 - Round[Count[te,0]/n1 ,.00001] \\
&del1 = edphi[y1], \\
&w1new = w1 + eta * Transpose[del1].train, \\
&w0new = w0 + eta * Total[del1]w0
\end{aligned}
$$

Iteration goes on till error becomes smaller than a given threshold.

4. RESULTS AND DISCUSSION

To demonstrate the robustness of the system and to justify forward propagation of untrained data samples, three experiments are conducted using secondary structure estimations of the tools given in Chou-Fasman website. The first experiment is made using Chou-Fasman estimates (C-F), the second by the use of Garnier-Osguthorpe-Robson (GOR) estimates, and finally the third by Neural Network estimate (ANN). Results from these experiments can be seen in Table 6.

	Training	Testing
CF	0.87260	0.85233
GOR	0.89800	0.89767
ANN	0.94860	0.92400
Average	90.64%	89.13%

Table 7 Performance measurements of three experiments using Chou-Fasman, GOR, and Neural Network correct estimates for the secondary structure.

If we analyze these results on the conformation type bases we observe highest correct estimate in α-helix, H; β-sheet E, and coil C.

	Correct Estimates %			
Estimator	C-F	GOR	ANN	Average
α-helix	90.73	93.20	92.48	92.14
β-sheet	88.40	83.79	75.77	81.92
Coil	66.67	85.07	94.01	82.65
Turn		81.03		81.03
Average	81.93	85.77	87.42	

Table 8 Correct estimates in α-helix, H; β-sheet E, and coil C.

5. CONCLUSIONS

Seven proteins are concatenated to create a sequence of 15536 residues. Then secondary structure of this sequence is obtained from Chou-Fasman web site. 10000 of these residues are used to train a simple perceptron with an input, and an output layer. Then the secondary structure of untouched 5536 residues with a success shown in Table 7 Mean rate of correct classification is around 90%, and quite satisfactory. We hope that the same success can be repeated using X-ray estimates of the second structures in training. It will be the topic of the next article.

REFERENCES

Chou, P. Y. $ Fasman, G. D. (1978). Advan. Enzymol. 47, 45-148.

Garnier, J., Osguthorpe, D. J. and Robson, B. (1978). J. Mol. Biol. 120, 97-120.

Hebb, B. O. (1949) The Organization of Behavior. New York: John Wiley & Sons. Introduction and Chapter 4 reprinted in Anderson & Rosenfeld, 1988, pp. 45-56.

Kabsch, W. T, and Sander, C. How good are predictions of protein secondary structure? (1983a). FEBS Letters, 155, 179-182.

Lim, V. I. (1974). J. Mol. Biol. 88, 873-894.

McCulloch, W. S. and Pill's, W. (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity." Bulletin of Mathematical Biophysics, 5:115-133. Reprinted in Anderson& Rosenfeld [1988], pp. 18-28.

Minsky, M. L. and Papert, S. A. (1988) Perceptrons, Expanded Edition. Cambridge, MA: MIT Press. Original edition, 1969.

Robson, B. and Pain, R. H. (1971). J. Mol. Biol. 58, 237-259.

Robson, B. and Suzuki, E. (1976). J. Mol. Biol. 107, 327-356.

Rosenblatt, E, (1958) The Perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review, vol. 65, pp. 386-408.

Taylor, W. R. (1986). J. Mol. Biol. 188, 233-258.

Vasquez M. , and Scheraga H.A. , 1985 Use of buildup and energy-minimization procedure to compute low-energy structures of the backbone of enkaphalin, Biopolymers 24:1437-1447.

Levitt, M., 1978. Conformational preferences of amino acids in globular proteins. Biochemistry 17, 4278–4285.

Webster, D., Gundersen, G. G.,Bulinski, J. C. and Borisy, G. G. . (1987a).Differential turnover of tyrosinated and detyrosinated microtubules. Proc. Natl. Acad. Sci. USA 84, 9040-9044.

H. Tang, K. C. Tan, and Z. Yi, Neural Networks: Computational Models and Applications, Springer-Verlag Berlin Heidelberg 2007.

Karplus, M., J.A. McCammon. Dynamics of Proteins: Elements and Function. In "Protein and Nucleic Acid Structure and Dynamics," J. King, Ed., Benjamin Cummings, Inc., pp. 169-206 (1985). (1985).