International Journal of Innovative Computing, Information and Control
Volume **14**, Number **3**, June **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 861–880

ANALYZING PROTEIN DATA USING UNSUPERVISED LEARNING TECHNIQUES

Silvana Albert, Mihai Teletin and Gabriela Czibula

Faculty of Mathematics and Computer Science
Babeş-Bolyai University
1, Mihail Kogălniceanu Street, Cluj-Napoca 400084, Romania
{ albert.silvana; gabis }@cs.ubbcluj.ro; tmic1334@scs.ubbcluj.ro

Received October 2017; revised February 2018

Abstract. *Proteins are the building blocks of life and the end result of the deoxyribonucleic acid decoding process. They make up the body of all living things and understanding fully their genesis and transformations is still the missing piece of the vast life puzzle. This paper investigates the usefulness of applying unsupervised machine learning methods for analyzing protein conformational transitions in order to extract meaningful information about their structural similarity. The structural similarity between proteins will be unsupervisedly uncovered using crisp and fuzzy self-organizing maps, based on proteins conformational transitions. We propose a method for modelling a protein based on its conformational transitions and we also examine how feature selection impacts the performance of the proposed models. The computational experiments performed on several protein data sets emphasize the effectiveness of using unsupervised learning models for capturing the similarity between proteins' structures. The obtained results also reveal that fuzzy models are able to increase the unsupervised model's performance.*

Keywords: Protein conformational transitions, Unsupervised learning, Self-organizing maps, Fuzzy self-organizing maps

1. **Introduction.**

Proteins have crucial roles in existence. They are at base, large, complex molecules that serve as building blocks in organisms (structural proteins) [1]. Another role is as catalyst in biochemical reactions in metabolisms (enzymes). They can also execute important tasks in order to maintain cellular environment. Proteins start folding immediately after they are synthesized and form a stable three-dimensional (3D) structure. When it comes to proteins, the 3D form dictates the biological function; thus the linear sequence of amino acids is the one that dictates the protein's purpose [2].

Not just the initial sequence of amino acids influences the protein's structures, we need to consider also the external factors located in the vicinity of the protein (like other molecules) and properties of the environment (like temperature) that can cause modifications in the protein's development. Proteins have a limited number of alternative conformations and can transition between them [3], and at any step, incorrect folding or mutations can occur. That is why understanding protein conformational transitions is extremely important for developing new better drugs that can inhibit the uncontrolled behaviour of a protein.

The protein's stable 3D structure is defined by a unique fold (topology) and the topology influences the folding mechanisms that are responsible for the protein's 3D structure [4]. However, the 3D structure is not stable since it keeps rearranging itself and transitioning to different structures, so it is safe to say that proteins are dynamic objects [3]. The structure dictates the biological function to be fulfilled by the protein. Some characteristics are similar in different proteins (like being hydrophobic) but there are also post-transitional modifications (like linking to iron atoms or sugars) that may alter the size and composition of the resulting protein. That is why we need to observe multiple proteins in determining what are the common parts that can be used in further learning and which are the particularities of a specific part of the protein that makes it unique.

The life cycle of a protein begins with the initial sequence of amino acids translated from the deoxyribonucleic acid. There are 20 amino acids that appear in the genetic code and they are building blocks that chain together in order to create a protein [5].

In a matter of seconds after a protein is synthesized, the intramolecular forces between the amino acids start the process of folding. The result of the folding process (which can be also described as a complex network formed by a set of elementary reactions [6]) is a stable three dimensional shape called the protein's native state.

The contribution of the paper is summarized as follows. Our main goal is to explore the usefulness of unsupervised learning models in identifying, starting from the conformational transitions of proteins, the structural relationships between them. We propose a vectorial representation of a protein which will be further used to build the unsupervised learning models. The crisp and fuzzy *self-organizing map* model will be comparatively analyzed and applied to several protein data sets. The obtained experimental results underline the effectiveness of using unsupervised learning models for capturing the similarity between proteins' structures and also reveal that the *fuzzy* models may have the potential to improve the learning performance. The study performed in this paper represents the starting point of a research which is being conducted in order to understand conformational changes in proteins, with the broader goal of learning to predict the conformational transitions of proteins. After studying the literature regarding the analysis of protein data we found out that a study similar to ours has not been performed, yet.

In this paper we seek answers to the following research questions:

(1) What is the potential of *fuzzy* self-organizing maps to unsupervisedly classify proteins according to their structural relationship? Are *fuzzy* self-organizing maps able to achieve a better mapping of proteins than the *crisp* self-organizing map (SOM)? In this direction, two case studies on protein data sets are conducted and the results provided by the non-fuzzy and fuzzy self-organizing maps will be comparatively presented.

(2) To what extent can feature selection improve the accuracy of identifying the structural relationship between proteins? We will analyze if selecting only relevant features would be helpful in increasing the performance of the unsupervised classification process.

The rest of the paper is organized as follows. The current state-of-the-art studies on modelling protein conformational transitions and protein structure analysis are presented in Section 2. The biological background on the problem of protein conformational changes is given in Section 3. The fundamental concepts regarding the machine learning models used are given in Section 4.1. With the goal of responding to our first research question, Section 4 introduces a *fuzzy* SOM model for uncovering the structural relationships between the proteins. The experimental evaluation is provided in Section 5 and Section 6 contains a discussion regarding the second research question we target. The conclusions of the paper and directions for future improvements are presented in Section 7.

2. **Literature Review.**

We review in the current section the existing approaches which are related to the modelling and analysis of conformational transitions and protein related data using computational intelligence methods.

Miyashita et al. [7], Whitford et al. [8], and Skjaerven et al. [9] have proposed different theoretical models for conformational transitions. These were used by physics-based computational methods, such as molecular dynamics [10] or Monte Carlo [11] to simulate the movement of atoms. These simulations have the potential to offer valuable information about protein structure, but they are extremely expensive from a computational viewpoint and thus their time intervals are considerably shorter than those of real biological conformational changes. Normal mode analysis [9] and simplifications of it have also been proposed for modelling protein conformational transitions: Schuyler et al. present in [12] a tool for generating a transition pathway from a source to a destination conformation and Al-Bluwi et al. use in [13] methods inspired from robotics (motion planning algorithms) to model conformational transitions.

Rao and Karplus investigate in [14] protein thermodynamics and dynamics through molecular dynamics (MD) simulations. The authors show that the inherent structures (IS) are able to provide an appropriate discretization of the trajectory and to reduce the shortcomings of previous approaches by using clustering. The IS are shown to facilitate the analysis of MD trajectories providing a decomposition of the conformation space which represents a natural and simple description of a dynamical system [14].

Self-organizing maps were proposed by Bouvier et al. in [15] as automatic tools for analyzing and clustering macromolecular conformations. A software package has been developed for exploiting the ability of SOMs to cluster macromolecular conformations. Using the U-matrix visualization, the authors showed that SOMs are useful for the analysis and visualization of the conformational space even when the conformations are spread over different areas. Besides, the map can also identify conformational changes such as catalytic mechanisms of an enzyme which are difficult to detect [15].

A study of the applicability of self-organizing maps for analyzing clusters formed by conformational ensembles of proteins is presented by Pandini et al. in [16]. The potential of using SOMs in computer-based approaches for medicinal chemistry is also highlighted through a drug-design experiment [16].

Self-organizing maps have been employed by Fraccalvieri et al. in [17] for analyzing molecular dynamics trajectories. Their approach combines SOM with hierarchical clustering for analyzing molecular structures. The output SOM prototype vectors are further used for linkage clustering and to decrease the computational cost of analysis, the geometric average was computed on properties of the sub groups of data. Finding optimal values was possible using the following parameters: the radius value was 3, the training length was 5000 and Gaussian was used for determining neighbours. They were able to compare multiple trajectories in order to determine how conformations flexibility can modulate domain functions.

The molecular dynamics of proteins is being analyzed using the previously described method in [18]. In this work the method is improved by introducing additional SOM analysis of specific mutants. Thus, the predictive potential of SOM, given proper data was demonstrated. The research described in [17, 18] highlighted the possibility of creating predictive tools for protein transitions.

Papaleo et al. described in [19] a computational method used to analyze the dynamics of myoglobin, a hemo-protein which is involved in oxygen transport and storage. Data was drawn from several molecular dynamics simulation. The analysis consisted of several methods: principal component analysis, free energy landscape analysis and clustering. The method was proven to be able to properly investigate the dynamical properties of the studied protein.

In [20] a method based on *fuzzy self-organizing maps* is employed in order to detect transmembrane segments of proteins. The data set composed of series of transmembrane proteins is classified using the proposed model into five classes. Some knowledge is obtained by analyzing the most frequent patterns. This is further used in order to identify and analyze such segments. The results show that the proposed model is powerful for correctly discriminating among segments.

Wang et al. propose in [21] a new architecture of radial basis neural network used to generate fuzzy classification rules. The results comparison showed that the fuzzy rules are outperforming other similar classification models such as decision trees and multi layered perceptrons. This work demonstrates the robustness of fuzzy rules over crisp models while analyzing protein sequences.

Fuzzy methods such as c-means clustering is used in [22] for clustering of gene expression data. The clustering algorithm is improved in order to enable the use of additional biological knowledge. The authors are militating that the capacity of assigning genes to multiple clusters in a fuzzy manner is much more appropriate than traditional clustering methods. The new method outperformed state-of-the-art models on two data sets. Moreover, the method was shown to be capable of producing biologically meaningful clusters.

The literature review presented in this section highlighted that SOMs and *fuzzy* SOMs are often used as unsupervised machine learning models for extracting meaningful information from conformational transitions and protein data. However, as far as we know, the fuzzy SOM models have not been applied yet for an unsupervised detection of structural similarities between proteins. Furthermore, we have not found a study similar to the analysis performed in this paper.

The SOM related literature contains several different approaches combining the *self-organizing maps* theory with the *fuzzy* sets theory developed by Zadeh [23]. A brief review of the existing FSOM approaches will be presented in the following.

A *fuzzy* Kohonen clustering network was introduced by Tsao et al. in [24], as a combination between the classical SOM model and the fuzzy c-means clustering (FCM) model. The hybrid approach was proposed as an optimization of FCM, the authors emphasizing that the method they proposed can be viewed as a Kohonen type of FCM [24]. The character of "self-organization" in the *fuzzy* Kohonen clustering network from [24] was given by the size of the updated neighborhood and the learning rate which were automatically adapted during the process of unsupervised learning.

Another approach combining *artificial neural networks* with fuzzy sets is the one from n [25]. A *fuzzy* SOM based on Kohonen's algorithm was introduced by Lei and Zheng [25]. In this approach, each node from the output map corresponds to a cluster and to a fuzzy set which was defined to characterize all input objects contained in that cluster. A difference between the approach from [25] and the classical SOM is that, instead of using a distance between an input object o and a neuron n from the map, it uses the membership of o to the cluster associated to neuron n. The authors argued that the proposed resulting method is able to deal with inexact or fuzzy information.

The problem of clustering relational data (i.e., containing a set of objects described by the distances between them is approached by Khalilia and Popescu in [26]. An algorithm called FRSOM was proposed by the authors by combining an extension of the SOM for handling relational data [27] with the fuzzy clustering algorithm extended for relational data [28]. The conclusion of the study from [26] was that FRSOM was able to detect substructures in the data which cannot be discovered by the crisp relational SOM.

Vuorimaa introduced in [29] a different perspective on a *fuzzy* self-organizing map, in which the map neurons were replaced by fuzzy rules. A classical SOM was used for learning the rules for each node from the output map. The map was designed to produce a single output value for each input instance. After the training of the map was completed, the rules from the neurons were used for providing weights to compute the final output for a new instance.

3. **Protein Structure Analysis.**

The foundation of life is considered to be the collection of large molecules called proteins. They build cells and have a huge role in how organisms develop, function, reproduce, feed and so on.

The alphabet representing amino acids consists of 20 letters, $\mathcal{A} = \{G, P, A, V, L, I, M, C, F, Y, W, H, K, R, Q, N, E, D, S, T\}$. The linear sequence of amino acids is called the *primary structure of a protein.*

There are millions of executed permutations going from the primary structure of a protein (the amino acid sequence), so it is not possible to predict, at this point, the 3D structure of a folded protein being given just the amino acids sequence. Illustrating the final form of a protein is a computationally difficult problem. Machine learning approaches could help solving this tricky task at hand by discovering hidden patterns in the multitude of permutations. Better understanding of the structure will give insights in predicting the function, thus being able to alter the undesired pathological effect.

Upon a more detailed evaluation, it was noticed that some conformations for small fragments were occurring frequently. They are called *states* and were encoded in structural alphabets (SA) [30]. Based on the various detection methods used, there are multiple types of structural alphabets that help facilitate the use of computational approaches in analyzing the protein structure. Basically, the 3D structure of a protein is represented by a one dimensional array containing the sequence of characters from the alphabet.

The structural alphabet derived by Pandini et al. in [30] will be used in our study. Each letter used in the representation describes a 3D short structural element determined by 4 amino acids from the linear sequence of the protein (primary structure).

There are currently multiple structural alphabets extracted and they can be evaluated in terms of being able to reconstruct unerringly the initial 3D protein structure. The assessment of the efficiency of the one we employed is available at [30].

There are 25 such letters in the structural alphabet $A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y$. It is important to mention that even if the same symbols are used both for amino acids and for structural elements, these are completely different concepts. In addition to these letters, a structural element is also characterized by the angles formed between consecutive amino acids. The alpha carbon atoms of the amino acids along with the torsion angle of all four atoms [30] are angles that preserve the 3 dimensional characteristics of the protein structure.

We will consider protein Pr and its primary sequence of amino acids $Pr = p_1p_2 \ldots p_n$. A sequence of letters of length $n - 3$ can be used for representing one of its structural conformations. The letters encode structures formed by four amino acids in the primary sequence $\mathcal{C} = s_1s_2 \ldots s_{n-3}$. Even a minor change in a protein's conformation will be a different representation.

The chosen protein for explaining conformations is 1HP9[1] which is a toxin found in scorpion venom. It has only 22 amino acids in its primary sequence: GHACYRNCWREGNDEETCKERC. Figure 1 depicts the three dimensional view of 1HP9. From the multiple possible SA representations available in [31] (database of molecular simulations), we listed five using the symbols of the structural alphabet [30].

- `QSUWNSVVVPRIJUUVVUV`
- `QSUWNSVVVPRIJUUVUUV`
- `RSUWNSVVVPRIKUUVVUV`
- `QSUWNSVVVPRGKUUVVUV`
- `QSUWNSVVVPRGKUUVVUV`

[1] http://www.rcsb.org/pdb/explore/explore.do?structureId=1hp9

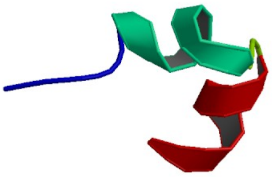


FIGURE 1. 3D view of protein 1HP9. Image from the RCSB PDB [32] of PDB ID 1HP9 [33] [2].

All the 5 representations have the same length: 19 symbols (initial sequence containing 22 amino acids -3 because 4 amino acids form a letter). It can be noticed that the differences between them are very small and in some parts of the sequence there is no difference at all. The time gap between these simulations is of 1 picosecond.

4. Methodology.

We introduce in the following the methodology our study is based on. We start by briefly describing in Section 4.1 the machine models that will be used in our approach. Then, the theoretical model which will be used in our approach is presented in Section 4.2. Section 4.3 introduces the *fuzzy self-organizing map* model which will be used in the experimental part of the paper (Section 5) for the unsupervised classification of proteins based on their structural relationships.

4.1. Machine learning models used.

From our data analysis perspective for the study of proteins conformations, we are going to use *self-organizing maps* and *principal component analysis* (PCA) as data visualization techniques. Furthermore, we will use 2 dimensional visualizations of the protein data sets in order to highlight similarities and differences between proteins and their conformations.

Self-organizing maps (SOMs) are *unsupervised* learning models related to the *artificial neural networks* literature, known to be powerful *data mining* tools for visualizing high-dimensional data. A *self-organizing map* [34] is a type of artificial neural network that is trained using *unsupervised learning* to provide a low-dimensional representation of a high-dimensional input space, called a *map* [35]. The main characteristic of SOMs is that the mapping between the input and the output space (map) preserves the *topology* of the input space.

Principal component analysis (PCA) [36] is an unsupervised learning method usually used to visualize high-dimensional data. The PCA algorithm encodes, using an ortogonal transformation, a function which maps the high dimensional input data points into a set of low dimensional output data points. This mapping is composed by *principal components* (PCs) which represent linearly uncorrelated variables which preserve the information encoded by the input data [36].

4.2. Theoretical model.

We have previously introduced in [37] a theoretical model for the problem of determining conformational transitions in proteins. A formalization was derived for the considered problem, starting from a data set of more than 300 proteins and their associated conformations [37]. We review in the following the modelling of a protein from [37] which will be also used in our study.

A protein Pr or length n is visualized as a word over the alphabet $\mathcal{A} = \{G, P, A, V, L, I, M, C, F, Y, W, H, K, R, Q, N, E, D, S, T\}$ of 20 letters representing amino acids: $Pr = p_1 p_2 \dots p_n$, where $p_i \in \mathcal{A}$, $\forall i \in \{1, 2, \dots, n\}$.

[2] This image is used according to RCSB PDB Policies & References: *http://www.rcsb.org/pdb/static.do?p=general_information/about_pdb/policies_references.html.*

Thousands of different conformations obtained by molecular dynamics simulations are given for a protein. Each conformation is converted into its SA representation. As shown in Section 3, the structural alphabet $\mathcal{SA}$ is composed of the 25 letters: $\mathcal{SA} = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y\}$.

Thus, for a protein Pr, we are given a large number m of experimentally determined conformations (for the data set we use, $m = 10000$). Therefore, for each protein we have a set of conformations, $\mathcal{S} = \left\{c_j \mid c_j = \left(c_j^1 c_j^2 \dots c_j^{n-3}\right),\ j \in \{1, 2, \dots, m\},\ c_j^k \in \mathcal{SA}\right\}$. Considering all these conformations, a distribution vector can be computed for each protein, which stores information about the SA elements' distribution in the protein's conformations. This frequency vector is constructed as follows. For each of the 25 letters from the structural alphabet $l_i \in \mathcal{SA}$ from the structural alphabet, we compute the probability $p_{l_i}^{Pr}$ of occurrence of letter l_i in the conformational transitions of protein Pr. Thus, a protein Pr may be visualized as a 25-dimensional vector containing the probabilities of occurrence of the symbols from the structural alphabet in the given protein, $Pr\left(p_{l_1}^{Pr}, p_{l_2}^{Pr}, \dots, p_{l_{25}}^{Pr}\right)$. For the protein example presented in Section 3 (1HP9), considering the 5 presented conformations, the frequency vector is presented in Table 1.

Table 1. Probabilities of occurrence of SA symbols for the example presented in Section 3

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**
0	0	0	0	0	0	0.021	0	0.031	0.021	0.031	0	0	0.052	0	0.052	0.042	0.063	0.105	0	0.221	0.305	0.052	0	0

4.3. **The proposed *fuzzy* self-organizing map.** We have previously proposed in [38] an algorithm for building a *fuzzy* self-organizing map (FSOM) which combines the classical SOM algorithm [34] with the concept of fuzziness from fuzzy clustering [39].

Considering the theoretical model introduced in Section 4.2, we assume that we have a set $\mathcal{P} = \{p_1, p_2, \dots, p_r\}$ of proteins represented as k-dimensional numerical vectors, i.e., $p_i = (p_{i1}, p_{i2}, \dots, p_{ik})$ (in our case the k is 25, as shown in Section 4.2). We will present in the following the FSOM algorithm for building, in an unsupervised manner, a two dimensional representation of $\mathcal{P}$ (i.e., the output map) by preserving the topology of the input protein space. The distance function used between the proteins is the *Euclidean distance* between their corresponding vectors.

The FSOM algorithm is based on the idea of using a *fuzzy membership* like in *fuzzy* clustering. Instead of having a single BMU for each instance, as in the classical SOM approach, the FSOM algorithm uses a membership matrix which specifies the degree to which an output neuron (which usually corresponds to a cluster) is the "winning neuron" for that input instance. Thus, instead of belonging to a single neuron (its BMU), an input instance will belong with some *membership degree* to all the neurons (clusters) from the map. Still, the main idea behind FSOM is that an input instance will have the maximum *membership degree* to the cluster (neuron) representing the "winner neuron" (BMU) from the crisp case. The weights updating rule idea is preserved from the classical SOM, but it also considers the membership values, such that a neuron to which an input instance has a larger membership will be "moved" closer to that instance than its neighbors [38].

We further consider that the input layer of the FSOM has k neurons, where k is the dimensionality of the input proteins (in our case $k = 25$). The map (output layer) consists of c neurons arranged on a two dimensional grid. In the FSOM approach, it is considered that each neuron from the computational layer represents a cluster; thus c has to be chosen equal or almost equal to the number of desired clusters. A neuron i from the input layer is connected to each neuron j from the output layer and the weight of this link is denoted by w_{ji}. Therefore, a neuron j from the output map may be viewed as a k-dimensional vector of weights, $w_j = (w_{j1}, w_{j2}, \ldots, w_{jk})$ [38].

We denote in the following by $u = (u_{ij})_{\substack{i=\overline{1,c} \\ j=\overline{1,r}}}$ the fuzzy *membership matrix* which describes a set of *fuzzy* c-partitions for the r input proteins. Thus, u_{ij} represents the degree to which protein p_j belongs to the output cluster (neuron) i.

The main idea of the FSOM algorithm is summarized below [38]. First, the weights are initialized with small random values ranging from 0 to 1. Then, the following steps are repeatedly performed until a stopping criterion is met (e.g., a given number of iterations):

1) The membership degrees values are computed as in Formula (1), where $m > 1$ is a real number (usually taken as 2), called the *fuzzifier*.

$$u_{il} = \frac{1}{\sum_{j=1}^{c} \left(\frac{||p_l - w_i||}{||p_l - w_j||} \right)^{\frac{2}{m-1}}} \tag{1}$$

2) Each input protein $p_t \in \mathcal{P}$ is fed to the map. First, the winning neuron j' from the output layer is determined as the neuron to which the protein has the maximum *membership degree*, i.e., $j' = \underset{1 \le j \le c}{\operatorname{argmax}}\ u_{jt}$. After the BMU was found, its weights as well as the weights of its neighbors are updated. More exactly, for the output neuron j ($\forall 1 \le j \le c$), its weights w_{ji} ($\forall 1 \le i \le k$) will be updated with a value Δw_{ji} given in Formula (2)

$$\Delta w_{ji} = \eta \cdot T_{jj'} \cdot (p_{ti} - w_{ji}) \cdot u_{jt}^{m} \tag{2}$$

where η represents the learning rate and $T_{jj'}$ is the neighborhood function that is generally employed in the classical Kohonen's algorithm [34].

The SOM related literature briefly presented in Section 2 highlighted that there are different ways to combine the theory of *self-organization* with the concept of *fuzziness*. Our FSOM proposed in this section for unsupervised classification of proteins based on their structural relationships, differs from the existing similar models at least by the way the BMU is computing and the method for updating the BMU's neighboring neurons.

5. Experimental Evaluation.

The goal of our experiments is to apply the *crisp* and *fuzzy self-organizing map* models to testing whether biologically relevant correlations could be unsupervisedly discovered within data sets of proteins. More exactly, we aim to investigate if the high dimensional representation of the proteins described in Section 4.2 is useful for uncovering structural relationships between the proteins.

Two protein data sets will be further used in our experiments. For a protein we will consider the vectorial representation based on the distributions of the symbols from the *structural alphabet* SA in the protein's conformations (see Section 4.2). Accordingly, a protein is visualized as a 25-dimensional vector containing the probabilities of occurrence for each SA symbol in the conformations of the given protein.

For the PCA analysis performed to obtain a two-dimensional view of the protein data (i.e., using the first two principal components having the highest explained variation ratio) we used the scikit-learn implementation of PCA available into the decomposition module [40].

5.1. Data sets.

We considered the structural alphabet representations of two data sets for our experiments obtained from the MoDEL database available at [31]. It includes data from multiple protein families and folding arguments.

Table 2 contains a description of the protein data sets used in our experiments. Each data set is composed by a number of protein *superfamilies*. For each data set, the second column in the table depicts the number of proteins in the data set, while the last column illustrates the number of superfamilies in the given data set. The superfamilies for the proteins were determined using **CATH Protein Structure Classification** database [41] which is a publicly available online resource that provides information on the evolutionary relationships of protein domains [42]. In this database, two proteins are considered in the same superfamily if there is a similarity between their three-dimensional structure [43].

Table 2. Description of used case studies

Data set	**# proteins**	**# superfamilies**
First	7	3
Second	57	9

5.1.1. *First data set.* Our first data set consists of *seven* proteins (codes: 1ASH, 1DLW, 1ECA, 1C52, 1CCR, 1APQ, 1COU in the Protein Data Bank [32]), taken from three different superfamilies (1.10.490.10, 1.10.760.10, 2.10.25.10). Table 3 illustrates the superfamilies for the seven proteins considered in our experiment, as well as the similarity index between the proteins belonging to the same superfamily, as provided by the FATCAT algorithm (Flexible structure AlignmenT by Chaining Aligned fragment pairs allowing Twists) [44].

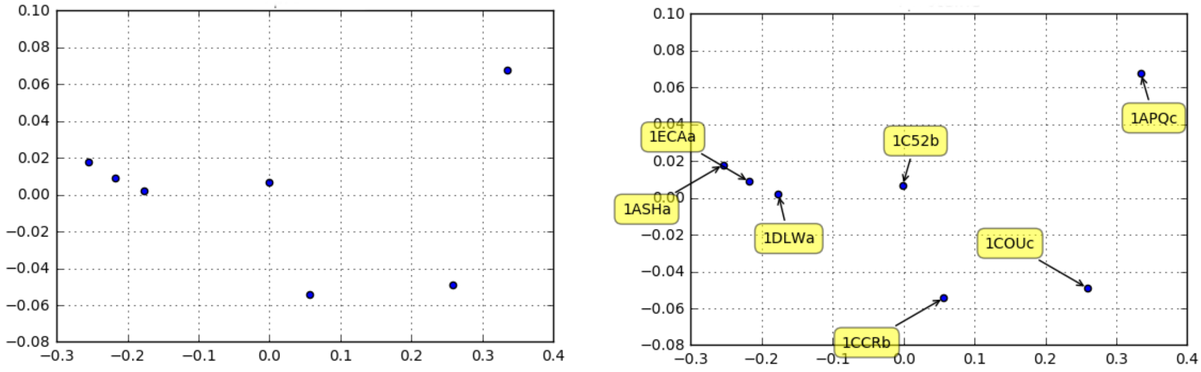
Table 3. Similarities between the proteins from the first data set

#	**Superfamily**	**Proteins**	**Similarity index**
1	**1.10.490.10**	{1ASH, 1DLW, 1ECA}	1ASH-1DLW: 20.57% 1ASH-1ECA: 25.85% 1ECA-1DLW: 19.08%
2	**1.10.760.10**	{1C52, 1CCR}	1C52-1CCR: 27.10%
3	**2.10.25.10**	{1APQ, 1COU}	1APQ-1COU: 4.92%

Figure 2 depicts a two-dimensional view of the first data set obtained using PCA. The data points are without labels in Figure 2(a) and labeled with the proteins names in Figure 2(b). The PCA graph reveals that similar proteins such as {1ASH, 1DLW, 1ECA} tend to cluster among each other, while proteins that have a lower similarity index such as {1APQ, 1COU} are rather distant on the plot. We may conclude that the data representation previously described in Section 4.2 is able to properly highlight the similarities and differences between different proteins.

5.1.2. *Second data set.* The second data set extends the set of proteins from Table 3. It consists of 58 proteins belonging to nine different families. The families names, as well as the proteins from each family are illustrated in Table 4.

Figure 3 depicts a two-dimensional view of the second data set obtained using PCA. The data points are without labels in Figure 3(a) and labeled with the family name in Figure 3(b). The PCA graph reveals again that, generally, proteins from the same families cluster together.

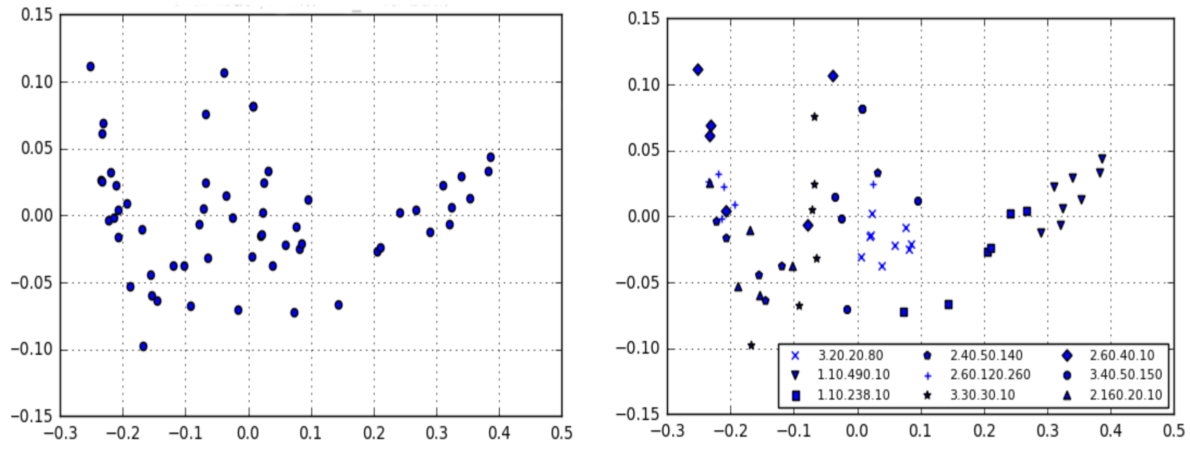


(a) (b)

FIGURE 2. PCA visualization for the proteins from the first data set

TABLE 4. Proteins from the second data set

#	Superfamily	Proteins
1	**3.20.20.80**	{1B1Y, 1CNV, 1EDG, 1ITX, 1JFX, 1KFW, 1NAR, 1VFF, 2EBN}
2	**1.10.490.10**	{1HLB, 1ITH, 1MBA, 2HBG, 2LHB, 1ASH, 1DLW, 1ECA}
3	**1.10.238.10**	{1OMR, 1SRA, 1UHN, 2SAS, 1CB1, 1IQ3}
4	**2.40.50.140**	{1SLJ, 1YVC, 1AH9, 1EOV, 1JT8, 1KRS}
5	**2.60.120.260**	{1NKG, 1PMJe 1ULO, 1GUI, 1I5P, 1K45}
6	**3.30.30.10**	{1PE4f, 1SEG, 1BCG, 1GPT, 1I2U, 1JXC}
7	**2.60.40.10**	{1R6V, 2FCB, 1JBJ, 1JE6, 1NCT, 1OLL}
8	**3.40.50.150**	{1Y8C, 1AF7, 1DUS, 1F3L, 1Y8C, 1YUB}
9	**2.160.20.10**	{1QCX, 1RU4, 1VBL, 1BHE, 1EE6}



(a) (b)

FIGURE 3. PCA visualization for the proteins from the second data set

5.2. **Evaluation measures.** In this section we present the external evaluation measures which will be used to estimate the performance of the SOM and FSOM applied on the protein data sets described in Section 5.1.

Let us assume that the data set under evaluation contains n proteins which belong to s superfamilies $\mathcal{F} = F_1, F_2, \ldots, F_s$ and the clusters (partitioning of the proteins) provided by the SOM/FSOM is composed by the set of clusters $\mathcal{K} = \{K_1, K_2, \ldots, K_s\}$.

We consider the generalized confusion matrix for our multiclass classification problem as $A = (a_{ij})_{\substack{i=\overline{1,s}\\ j=\overline{1,s}}}$, where a_{ij} is computed as the number of proteins predicted as belonging to class (superfamily) i (K_i) and having the actual superfamily j (F_j).

The *precision* $Prec_i$ of cluster K_i, $\forall 1 \le i \le s$ is defined as the maximum number of proteins from K_i correctly assigned in a superfamily, $Prec_i = \frac{\max\limits_{j=1,s} a_{ij}}{\sum_{j=1}^{s} a_{ij}}$. The *recall* of the i-th superfamily F_i ($\forall 1 \le i \le s$), denoted by Rec_i is computed as the maximum number of proteins from a cluster which were correctly placed in the superfamily F_i, i.e., $Rec_i = \frac{\max\limits_{j=1,s} a_{ji}}{\sum_{j=1}^{s} a_{ji}}$.

The overall *precision* value (**Precision**) for the output partition $\mathcal{K}$ is computed as the average of the precision values for all s clusters obtained, i.e., $Precision = \frac{\sum_{i=1}^{s} Prec_i}{s}$. The overall *recall* (**Recall**) value for the set $\mathcal{F}$ of superfamilies with respect to the output partition $\mathcal{K}$ is computed as the average of the recall values for all s superfamilies, i.e., $Recall = \frac{\sum_{i=1}^{s} Rec_i}{s}$.

The *F-measure* (**F-measure**) for the obtained partition $\mathcal{K}$ is computed as the harmonic mean between *precision* and *recall*, i.e.,

$$F\text{-}measure = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

The *precision*, *recall* and *F-measure* take values from 0 to 1 and higher values indicate better partitions, i.e., a better performance for the classification process.

5.3. **Results.** In order to test the potential of self-organizing maps in unsupervisedly classifying the proteins according to their structural relationship, a *crisp* and a *fuzzy* SOM experiment will be performed on each of the data sets described in Section 5.1. Our goal is to compare the performance of the crisp and *fuzzy* SOMs in detecting proteins which are similar regarding their three-dimensional structure.

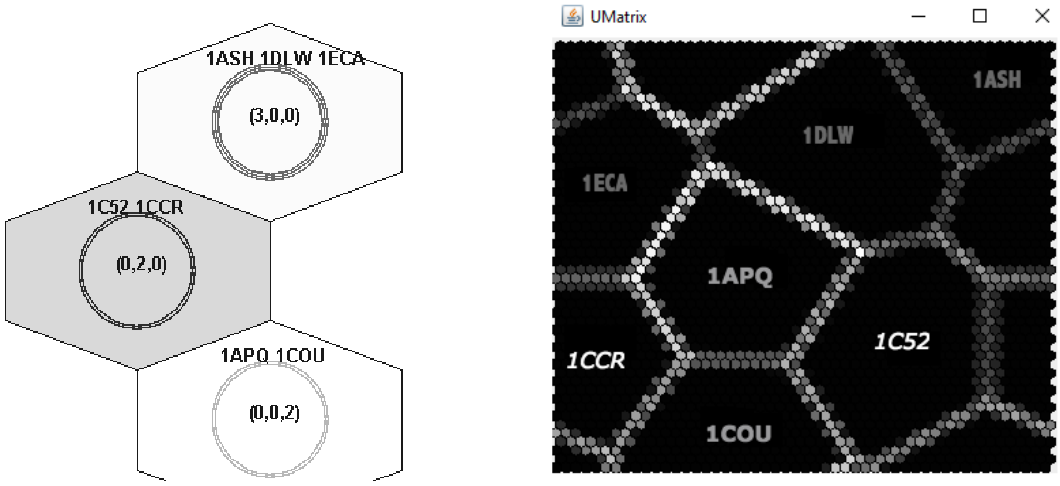
Considering the proteins' modelling from Section 4.2, we map the proteins (considering their 25-dimensional representations) from the considered data sets on an SOM (respectively an FSOM) having a *torus* topology. For the visualization of the maps, we use the U-Matrix method [45] with the following interpretation: the lighter regions express data that are dissimilar while darker regions contain data that are similar. For training the maps, we used 200000 iterations and a learning rate of 0.1.

For each experiment, we tested different configurations for the SOM. For the FSOM, the theoretical studies indicate that a number of neurons approximately equal to the number of clusters have to be used. Therefore, because we aim to provide a more accurate comparison between the performance of the SOM and FSOM, we used the same configurations for both maps. On each neuron from both the SOM and FSOM, the proteins which are mapped on that neuron (i.e., the neuron is its BMU) are represented as circles. In addition, on each neuron we represent a vector whose elements specify how many proteins from each superfamily are mapped on that neuron.

During our experiments we observed that the topology of the output space (i.e., the neighborhood between the neurons from the map) is preserved regardless of the dimension of the map. As depicted in Figures 4(b) and 6(b), the U-matrix for a larger SOM configuration may be viewed as zooming in (extending) the U-matrix for a smaller SOM configuration (Figures 4(a) and 6(a)).

Figure 4 illustrates the SOM trained on the seven proteins from the first data set. In the left side image a configuration with 3 neurons is used, since 3 superfamilies (clusters) are present in the data set. In the right side image a larger configuration for the SOM (50×50 neurons) was used. For clarity reasons, the name of the proteins which are mapped on each neuron is also represented. We observe that a perfect mapping was obtained by the 3×1 SOM (Figure 4(a)), namely each neuron contains the proteins from a superfamily. The same accurate mapping may be seen in Figure 4(b) where we clearly observe on the map three regions corresponding to the three protein superfamilies.

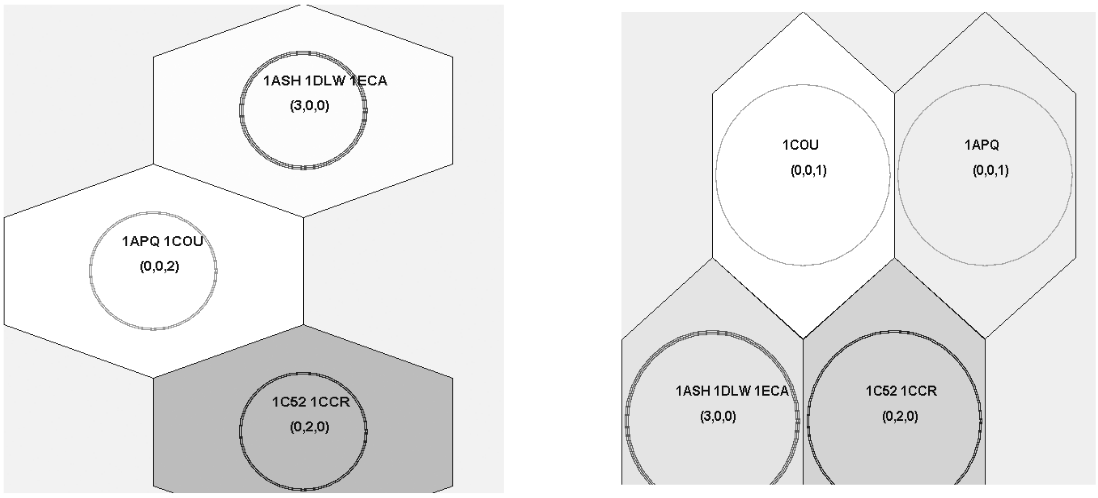
Figure 5 depicts the FSOM trained on the proteins from the first data set. In the left side image a configuration with 3 neurons (i.e., the number of superfamilies) is used, while in the right side image a configuration with 4 neurons (2×2 neurons) is used for the FSOM. On both maps each protein was perfectly classified in its superfamily. On the FSOM with 2×2 neurons we observe that the proteins 1COU and 1APQ were mapped



(a) 3×1 neurons

(b) 50×50 neurons

Figure 4. U-Matrix for the SOM trained on the proteins from the first data set



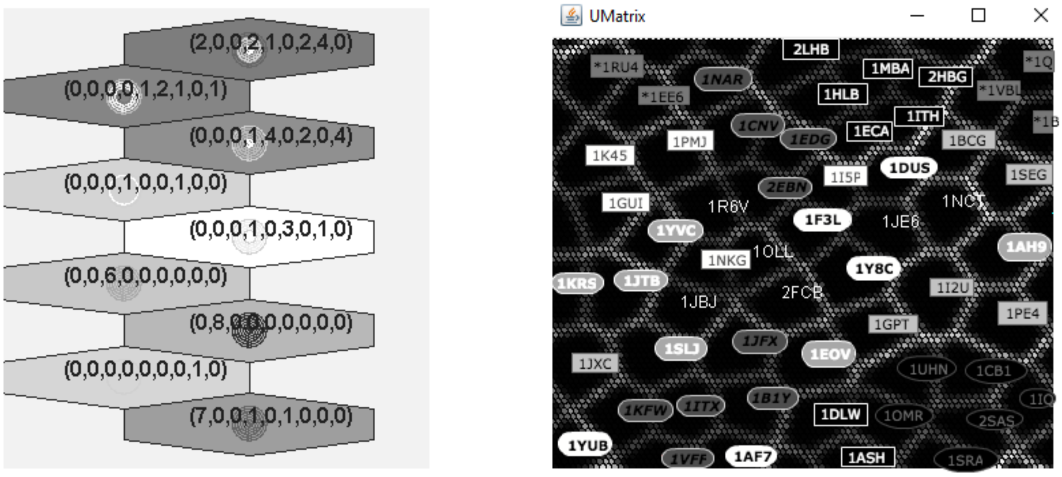
(a) 3×1 neurons

(b) 2×2 neurons

Figure 5. U-Matrix for the FSOM trained on the proteins from the first data set

on the neighboring neurons from the first row of the map. This mapping is explainable, since as shown in Table 3 the similarity between proteins 1COU and 1APQ is only 4.92%.

The U-Matrix for the SOM trained on the 57 proteins from the second data set is shown in Figure 6. In the left side image a configuration with 9 neurons is used, since 9 superfamilies (clusters) exist in the data set. In the right side image a larger configuration for the SOM (85 × 85 neurons) was used. On each neuron we represent a vector whose elements specify how many proteins from each superfamily are mapped on that neuron. We observe in Figure 6(a), that there are several misclassifications, namely there are neurons (corresponding to clusters) which contain proteins from different superfamilies. Only three superfamilies were perfectly identified. An extended view of the mapping from Figure 6(a) is observed in Figure 6(b), where the misclassifications are better observed on a larger map.



(a) 9 × 1 neurons

(b) 85 × 85 neurons

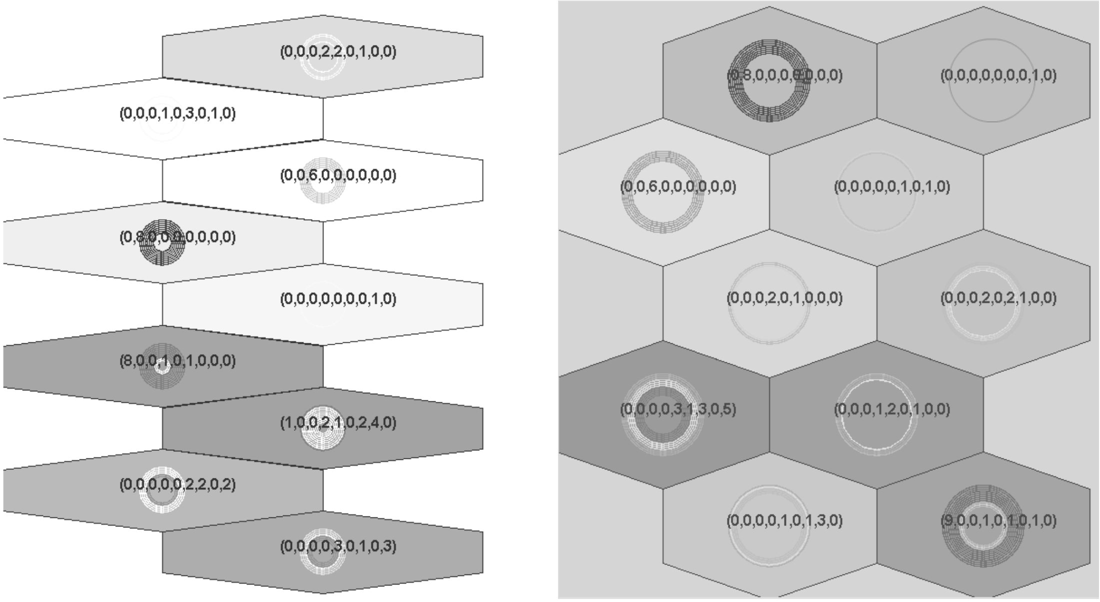
Figure 6. U-Matrix for the SOM trained on the proteins from the second data set

The FSOM trained on the proteins from the second data set is illustrated in Figure 7. In the left side image a configuration with 9 neurons (i.e., the number of superfamilies from the data set) is used, whereas in the right side image a configuration with 10 neurons (5 × 2 neurons) is used for the FSOM. As for the crisp case, the FSOMs from Figure 7(a) show several misclassifications. One observes that in the FSOM from Figure 7(b) the two neighboring neurons from the third row of the map may be viewed as subclusters of the fourth superfamily.

For the SOMs and FSOMs depicted in Figures 4, 5, 6 and 7 we present in Table 5 the values for the three evaluation measures (*precision*, *recall* and *f-measure*) computed as described in Section 5.2.

From Table 5 we observe that on the first data set, both the SOM and FSOM have perfectly classified the proteins in superfamilies. For the second data set, the FSOM has a slightly better performance on the configuration with 10 neurons (5 × 2 neurons) from Figure 7(b). We remark that the poorer performance of both SOM and FSOM on the second data set with 57 proteins may be due to the vectorial representation of the proteins which only considers the distributions of the SA symbols.

6. **Discussion.** In this section we will focus on investigating the second research question formulated at the beginning of our paper. More specifically, we analyze if selecting only



(a) 9 × 1 neurons

(b) 5 × 2 neurons

Figure 7. U-Matrix for the FSOM trained on the proteins from the second data set

Table 5. Comparative performance of SOM and FSOM

Data set	**Model**	**Map configurations**	**Precision**	**Recall**	**F-measure**
First	SOM	3 × 1	1	1	1
	FSOM	3 × 1	1	1	1
	FSOM	2 × 2	1	1	1
Second	SOM	9 × 1	0.667	0.675	0.671
	FSOM	9 × 1	0.662	0.647	0.654
	FSOM	5 × 2	0.696	0.741	**0.718**

relevant letters from the structural alphabet can improve the accuracy of identifying the structural relationship between proteins.

For testing if feature selection can improve the accuracy of identifying the structural relationship between proteins, we perform an experiment on both proteins data sets described in Section 5.1.2. We want to analyze if, in the vectorial representation of the proteins, selecting only relevant letters from the structural alphabet, instead of all the symbols, would be helpful in increasing the performance of the unsupervised classification process.

We remind that the vectorial representation of the proteins used so far in the paper is a *distribution* based representation introduced in Section 4.2. In this representation, a protein is expressed as a 25-dimensional numerical vector containing the probabilities of occurrence of the symbols from the structural alphabet $\mathcal{SA}$ in the given protein.

For identifying the relevance of the features (SA symbols) in the classification process, we computed the R^2 values between the features and the known output class (i.e., superfamily to which a protein belongs). These values are graphically represented in Figure 8 for the first data set and in Figure 11 for the second data set. The R^2 (*R-squared*) value between two random variables X and Y is a statistical measure computed as the square of the *Pearson correlation* [46] between X and Y. Its value varies from 0.0 to 1.0, the

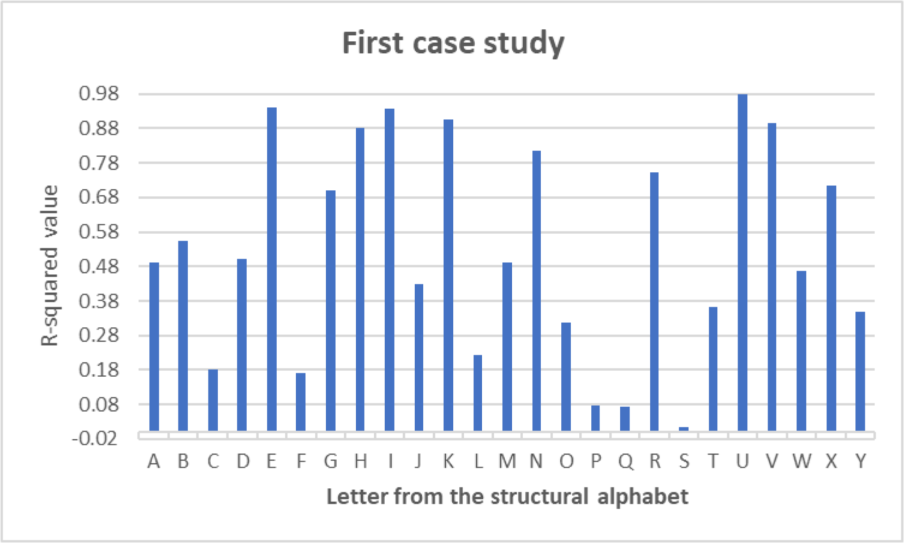
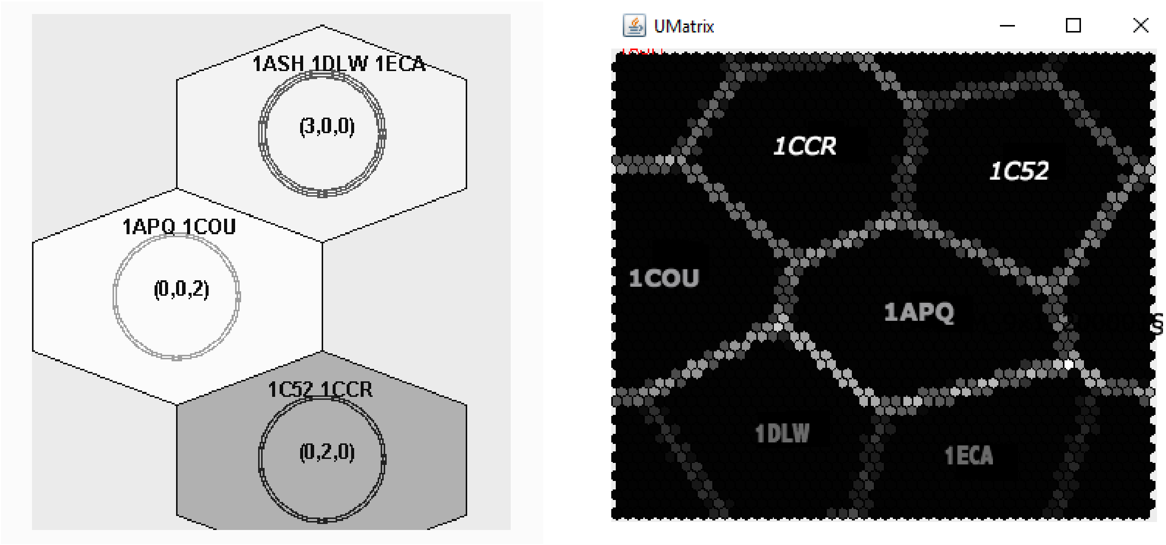


Figure 8. Visualization of the R^2 values between the features and the output class for the first data set



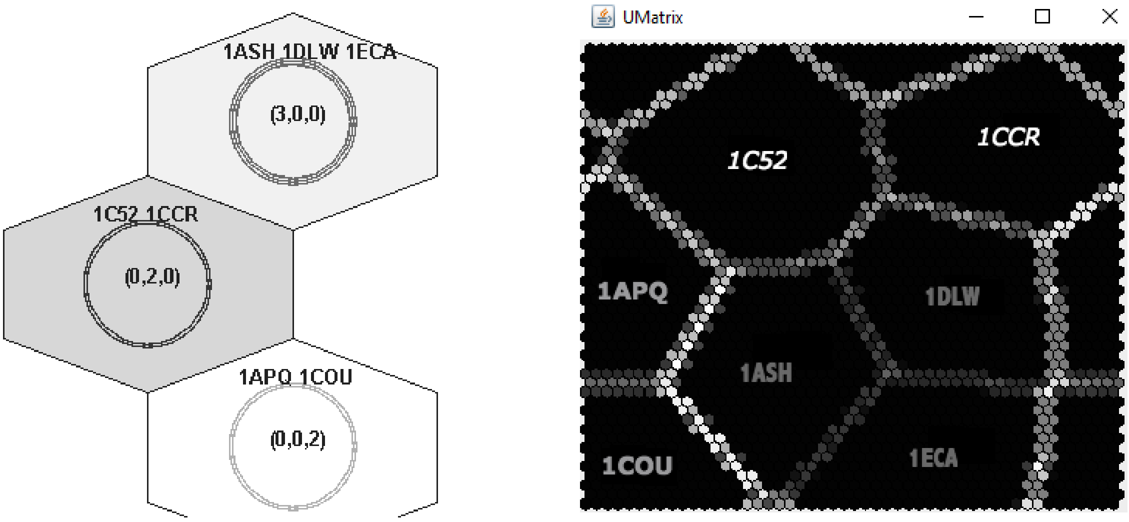
(a) 3 × 1 neurons

(b) 50 × 50 neurons

Figure 9. U-Matrix for the SOM trained on the proteins from the first data set, after removing features C, F, P and S. The threshold is 0.2.

higher values being preferable, since they express a higher linear correlation between the random variables.

Analyzing the R^2 values from Figure 8 we observe that there are four features for which the R^2 values are small, below 0.2. These symbols are: C ($R^2 = 0.18$), F ($R^2 = 0.17$), P ($R^2 = 0.076$) and S ($R^2 = 0.013$). We decided to use two thresholds for removing the features: 0.2 and 0.1. Figure 9 depicts the U-Matrix for the SOM trained on the 21-dimensional proteins from the first data set (the symbols C, F, P and S were removed from the feature set). In the left side image a configuration with 3 neurons (i.e., the number of superfamilies) is used, while in the right side image a larger configuration with 50 × 50 neurons is used for the SOM.



(a) 3 × 1 neurons

(b) 50 × 50 neurons

Figure 10. U-Matrix for the SOM trained on the proteins from the first data set, after removing features P and S. The threshold is 0.1.

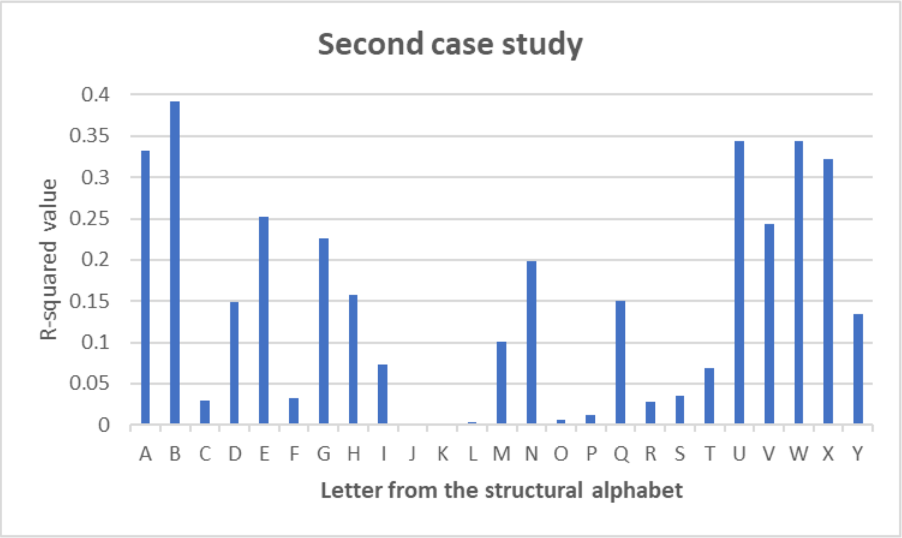
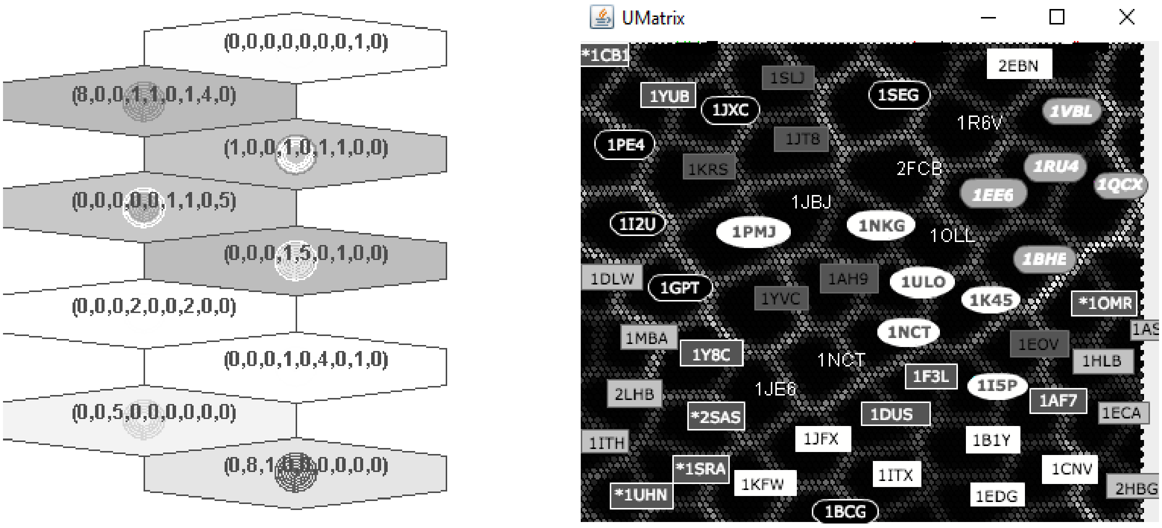


Figure 11. Visualization of the R^2 values between the features and the output class for the second protein data set

Figure 10 depicts the U-Matrix for the SOM trained on the 23-dimensional proteins from the first data set (the symbols P and S were removed from the feature set). In the left side image a configuration with 3 neurons (i.e., the number of superfamilies) is used, while in the right side image a larger configuration with 50 × 50 neurons is used for the SOM.

From Figures 9 and 10 we observe that, on the first protein data set, the feature selection step has no influence on the performance of the SOM in classifying the proteins according to their similarity.

Analyzing the R^2 values from Figure 11 we observe that there are four features for which the R^2 value between them and the output class is very small, below 0.01. These



(a) 9 × 1 neurons

(b) 85 × 85 neurons

FIGURE 12. U-Matrix for the SOM trained on the proteins from the second data set, after feature selection. The threshold is 0.01.

TABLE 6. Impact of feature selection on SOM trained on the first and second data sets. Different values for the threshold τ are considered.

Data set	Threshold τ	Removed letters from the structural alphabet	Precision	Recall	F-measure
First	–	None	1	1	1
	0.2	$\{C, F, P, S\}$	1	1	1
	0.1	$\{P, S\}$	1	1	1
	0.01	None	1	1	**1**
Second	–	None	0.667	0.675	0.671
	0.01	$\{J, K, L, O\}$	0.697	0.728	**0.712**

symbols are: J ($R^2 = 0.0008$), K ($R^2 = 0.0008$), L ($R^2 = 0.004$) and O ($R^2 = 0.007$). We decided to use 0.01 the threshold for removing the features.

Figure 12 depicts the U-Matrix for the SOM trained on the 21-dimensional proteins from the second data set (the symbols J, K, L and O were removed from the feature set). In the left side image a configuration with 9 neurons (i.e., the number of superfamilies) is used, while in the right side image a larger configuration with 85 × 85 neurons is used for the SOM.

In order to evaluate the impact of feature selection on the proteins classification process, we comparatively present in Table 6 the values for the evaluation measures (*precision*, *recall* and *f-measure*) computed as described in Section 5.2 for the SOM on the two data sets with all features and with removing features whose R^2 value is below a threshold τ. For computing the evaluation measures we considered the SOMs from Figures 9(a), 10(a) and 12(a).

Analyzing the results from Table 6 we remark that the feature selection step improved a little the performance of the classification for the second data set containing 57 proteins. On this data set, the values for *precision*, *recall* and *F-measure* are slightly better when the feature selection step was applied.

Still, we consider that the results presented above may be influenced by the vectorial representation used for the proteins which captures only quantitative information and thus might not be relevant enough.

7. **Conclusions and Future Work.** We have conducted in this paper a study towards the application of *unsupervised machine learning* methods for analyzing protein conformational transitions in order to extract information about their structural similarity. Through several experiments conducted on three protein data sets, we investigated the usefulness of *fuzzy* self organizing maps in identifying the structural relationship between proteins.

Future work will be done in order to extend the experimental evaluation performed in this paper on other larger protein data sets in order to obtain a better validation for the above mentioned conclusions of our study.

Acknowledgment. The authors thank lecturer Alessandro Pandini from Brunel University, London for providing the protein data sets used in the experiments and also acknowledge the assistance received from lecturer Iuliana Bocicor from Babeş-Bolyai University, Romania for the feedback regarding the paper and the computational results. Many thanks are due to the editor and the anonymous reviewers, for their helpful comments and suggestions which have improved the paper and the presentation.

REFERENCES

[1] A. Lesk, *Introduction to Protein Science*, Oxford, 2004.

[2] D. Voet and J. Voet, *Biochemistry*, 4th Edition, Wiley, 2011.

[3] N. Tokuriki and D. S. Tawfik, Protein dynamism and evolvability, *Science*, vol.324, no.9524, pp.203-207, 2009.

[4] D. Baker, A surprising simplicity to protein folding, *Nature*, vol.405, no.6782, pp.39-42, 2000.

[5] A. Ambrogelly, S. Palioura and D. Söll, Natural expansion of the genetic code, *Nature Chemical Biology*, vol.3, pp.29-35, 2007.

[6] J. N. Onuchic and P. G. Wolynes, Theory of protein folding, *Current Opinion in Structural Biology*, vol.14, no.1, pp.70-75, 2004.

[7] O. Miyashita, P. G. Wolynes and J. N. Onuchic, Simple energy landscape model for the kinetics of functional transitions in proteins, *The Journal of Physical Chemistry B*, vol.109, no.5, pp.1959-1969, 2005.

[8] P. C. Whitford, O. Miyashita, Y. Levy and J. N. Onuchic, Conformational transitions of adenylate kinase: Switching by cracking, *Journal of Molecular Biology*, vol.366, no.5, pp.1661-1671, 2007.

[9] L. Skjaerven, S. M. Hollup and N. Reuter, Normal mode analysis for proteins, *Journal of Molecular Structure: THEOCHEM*, vol.898, nos.1-3, pp.42-48, 2009.

[10] K. Okazaki, N. Koga, S. Takada, J. N. Onuchic and P. G. Wolynes, Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: Structure-based molecular dynamics simulations, *Proc. of the National Academy of Sciences*, vol.103, no.32, pp.11844-11849, 2006.

[11] I. Lotan, F. Schwarzer and J. C. Latombe, Efficient energy computation for Monte Carlo simulation of proteins, *Lecture Notes in Computer Science*, vol.2812, pp.354-373, 2003.

[12] A. D. Schuyler, R. L. Jernigan, P. K. Qasba, B. Ramakrishnan and G. S. Chirikjian, Iterative cluster-nma: A tool for generating conformational transitions in proteins, *Proteins: Structure, Function, and Bioinformatics*, vol.74, no.3, pp.760-776, 2009.

[13] I. Al-Bluwi, M. Vaisset, T. Siméon and J. Cortés, Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods, *BMC Structural Biology*, vol.13, no.1, 2013.

[14] F. Rao and M. Karplus, Protein dynamics investigated by inherent structure analysis, *Proc. of the National Academy of Sciences*, vol.107, no.20, pp.9152-9157, 2010.

[15] G. Bouvier, N. Desdouits, M. Ferber, A. Blondel and M. Nilges, An automatic tool to analyze and cluster macromolecular conformations based on self-organizing maps, *BMC Bioinformatics*, vol.31, no.9, pp.1490-1492, 2015.

[16] A. Pandini, D. Fraccalvieri and L. Bonati, Artificial neural networks for efficient clustering of conformational ensembles and their potential for medicinal chemistry, *Current Topics in Medicinal Chemistry*, vol.13, pp.642-651, 2013.

[17] D. Fraccalvieri, A. Pandini, F. Stella and L. Bonati, Conformational and functional analysis of molecular dynamics trajectories by self-organising maps, *BMC Bioinformatics*, vol.12, 2011.

[18] D. Fraccalvieri, M. Tiberti, A. Pandini, L. Bonati and E. Papaleo, Functional annotation of the mesophilic-like character of mutants in a cold-adapted enzyme by self-organising map analysis of their molecular dynamics, *Mol. BioSyst.*, vol.8, pp.2680-2691, 2012.

[19] E. Papaleo, P. Mereghetti, P. Fantucci, R. Grandori and L. D. Gioia, Free-energy landscape, principal component analysis, and structural clustering to identify representative conformations from molecular dynamics simulations: The myoglobin case, *Journal of Molecular Graphics and Modelling*, vol.27, no.8, pp.889-899, 2009.

[20] Y. Deng, TSFSOM: Transmembrane segments prediction by fuzzy self-organizing map, *Advances in Neural Networks – ISNN*, pp.728-733, 2006.

[21] D. Wang, N. K. Lee, T. S. Dillon et al., Extraction and optimization of fuzzy protein sequences classification rules using GRBF neural networks, *Neural Information Processing – Letters and Reviews*, vol.1, no.1, pp.53-57, 2003.

[22] L. Tari, C. Baral and S. Kim, Fuzzy c-means clustering with prior biological knowledge, *Journal of Biomedical Informatics*, vol.42, no.1, pp.74-81, 2009.

[23] L. A. Zadeh, A summary and update of "fuzzy logic", *IEEE International Conference on Granular Computing*, San Jose, CA, USA, pp.42-44, 2010.

[24] E. C.-K. Tsao, J. C. Bezdek and N. R. Pal, Fuzzy Kohonen clustering networks, *Pattern Recognition*, vol.27, no.5, pp.757-764, 1994.

[25] P. Lei and H. Zheng, Clustering properties of fuzzy Kohonen's self-organizing feature maps, *Journal of Electronics*, vol.12, no.2, pp.124-133, 1995.

[26] M. Khalilia and M. Popescu, Fuzzy relational self-organizing maps, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp.1-6, 2012.

[27] A. Hasenfuss and B. Hammer, Relational topographic maps, in *Advances in Intelligent Data Analysis VII, Proc. of the 7th International Symposium on Intelligent Data Analysis*, M. R. Berthold, J. Shawe-Taylor and N. Lavrac (eds.), vol.4723, 2007.

[28] R. J. Hathaway and J. C. Bezdek, Nerf c-means: Non-Euclidean relational fuzzy clustering, *Pattern Recognition*, vol.27, no.3, pp.429-437, 1994.

[29] P. Vuorimaa, Fuzzy self-organizing map, *Fuzzy Sets and Systems*, vol.66, pp.223-231, 1994.

[30] A. Pandini, A. Fornili and J. Kleinjung, Structural alphabets derived from attractors in conformational space, *BMC Bioinformatics*, vol.11, no.97, pp.1-18, 2010.

[31] T. Meyer, M. D'Abramo, A. Hospital, M. Rueda, C. Ferrer-Costa, A. Pérez, O. Carrillo, J. Camps, C. Fenollosa, D. Repchevsky, J. L. Gelpí and M. Orozco, MoDEL (molecular dynamics extended library): A database of atomistic molecular dynamics trajectories, *Structure*, vol.18, no.11, pp.1399-1409, 2010.

[32] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, The protein data bank, *Nucleic Acids Research*, vol.28, pp.235-242, 2000.

[33] K. N. Srinivasan, V. Sivaraja, I. Huys, T. Sasaki, B. Cheng, T. K. Kumar, K. Sato, J. Tytgat, C. Yu, B. C. San, S. Ranganathan, H. J. Bowie, R. M. Kini and P. Gopalakrishnakone, kappa-Hefutoxin1, a novel toxin from the scorpion heterometrus fulvipes with unique structure and function. Importance of the functional diad in potassium channel selectivity, *J. Biol. Chem.*, vol.277, pp.30040-30047, 2002.

[34] P. Somervuo and T. Kohonen, Self-organizing maps and learning vector quantization for feature sequences, *Neural Processing Letters*, vol.10, pp.151-159, 1999.

[35] N. Elfelly, J.-Y. Dieulot and P. Borne, A neural approach of multimodel representation of complex processes, *International Journal of Computers, Communications & Control*, vol.III, no.2, pp.149-160, 2008.

[36] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

[37] A. Pandini, M.-I. Bocicor, G. Czibula, S. Albert and M. Teletin, Using computational intelligence models for additional insight into protein structure, *Studia Universitatis Babes-Bolyai Informatica*, vol.62, pp.107-119, 2017.

[38] I.-G. Czibula, G. Czibula, Z. Marian and V.-S. Ionescu, A novel approach using self-organizing maps for detecting software faults, *Studies in Informatics and Control*, vol.25, no.2, pp.207-216, 2016.

[39] F. Klawonn and F. Höppner, What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier, *Lecture Notes in Computer Science*, vol.2810, pp.254-264, 2003.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research*, vol.12, pp.2825-2830, 2011.

[41] CATH: Protein structure classification database at UCL, *CATH/Gene3D*, http://www.cathdb.info.

[42] N. L. Dawson, T. E. Lewis, S. Das, J. G. Lees, D. Lee, P. Ashford, C. A. Orengo and I. Sillitoe, CATH: An expanded resource to predict protein function through structure and sequence, *Nucleic Acids Research*, vol.45, no.D1, pp.D289-D295, 2017.

[43] M. Knudsen and C. Wiuf, The CATH database, *Human Genomics*, vol.4, pp.207-212, 2010.

[44] Y. Ye and A. Godzik, FATCAT: A web server for flexible structure comparison and structure similarity searching, *Nucleic Acids Research*, vol.32, pp.582-585, 2004.

[45] S. Kaski and T. Kohonen, Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world, *Neural Networks in Financial Engineering. Proc. of the 3rd International Conference on Neural Networks in the Capital Markets*, pp.498-507, 1996.

[46] S. Tufféry, *Data Mining and Statistics for Decision Making*, John Wiley and Sons, 2011.