CLUSTER VISUALIZATION AND NONLINEAR PROJECTION TECHNIQUES FOR BIOLOGICAL SEQUENCES

C. Ferles[*], *A. Stafylopatis*[†]

Abstract: The present study devises two techniques for visualizing biological sequence data clusterings. The Sequence Data Density Display (SDDD) and Sequence Likelihood Projection (SLP) visualizations represent the input symbolical sequences in a lower-dimensional space in such a way that the clusters and relations of data elements are preserved as faithfully as possible. The resulting unified framework incorporates directly raw symbolical sequence data (without necessitating any preprocessing stage), and moreover, operates on a pure unsupervised basis under complete absence of prior information and domain knowledge.

Key words: *non-numerical/symbolical sequence data, DNA, RNA, protein, clustering, visualization, mapping, nonlinear projection, Self-Organizing Map (SOM), Hidden Markov Model (HMM)*

Received: May 27, 2014 **DOI:** 10.14311/NNW.2016.26.016
Revised and accepted: May 2, 2016

1. Introduction

Deoxyribonucleic acid (DNA), ribonucleic acid (RNA) and chain molecules are characteristic paradigms of sequence data which are intrinsically encoded in either the four-letter alphabet of nucleotides or the twenty-letter alphabet of amino acids. As a result, machine learning approaches that are in position to incorporate and process such symbolical sequence data have proven effective for modeling, processing and analyzing biological molecules. On the contrary, machine learning techniques that are able to process numerical data can be employed only after interposing a preprocessing stage in order to transform symbolical sequences to numerical data spaces. Nevertheless, additional computational complexity, and frequently, loss of information are inevitable aftereffects of such preprocessing transformations.

One common situation in early stages of bioinformatics projects is that the only available information comes in the form of symbolical sequence data (viz. DNA, RNA and protein sequences); any other kind of prior information or domain

[*] Christos Ferles – Corresponding author, Intelligent Systems Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Zografou, Athens, Greece, E-mail: `christos.ferles@gmail.com`

[†] Andreas Stafylopatis, Intelligent Systems Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Zografou, Athens, Greece

knowledge is, or is considered to be, inexistent. Clustering's objective is to produce simplified descriptions and summaries of large data sets by adopting primordial exploration strategies which necessitate little or no prior knowledge. Clustering [5, 28] is one standard method that can create higher abstractions (symbolisms) from raw data automatically; another alternative method is to project high-dimensional data as points on a low-dimensional display. Because the Self-Organizing Map (SOM) [14] is a combination of these two methods, it has great capabilities as a data clustering, mapping and visualization algorithm.

In particular, the connectivity strength matrix visualization scheme [25] uses a topology representing graph that projects local data distributions/distances within receptive fields. These graphs are merged in subsequent stages so as to achieve improved visualizations. The approach in [4] suggests an enhanced version of the Clusot algorithm for automatic cluster detection on the SOM. In essence, the Clusot [3] is a two-step procedure, where the computation of an appropriate surface is used subsequently for cluster detection. The first phase of this procedure is a more elaborate way to visualize the information which is captured by the SOM when compared to other well-known approaches like the U-matrix [26]. Similarly, in [16] the inter-point distances in the feature space between the SOM neurons are used for presenting graphically the underlying structure of the data. A further refinement of these approaches is the visualization-induced SOM [29] an algorithm that regularizes and scales the inter-neuron distances so as to control the resolution of the mappings.

The Self-Organizing Hidden Markov Model Map (SOHMMM) [8, 11] is a hybrid unsupervised approach that can process and analyze DNA, RNA [11], protein chain molecules [7, 10], and generic sequences [9] of high dimensionality and variable lengths encoded directly in non-numerical/symbolical alphabets. On the contrary, various other models that extend the SOM for processing sequence data are in position to incorporate and analyze symbolical sequences only after an intermediate preprocessing stage (e.g. orthogonal/Euclidean encoding, weighted Levenshtein/local feature distances, pairwise dissimilarities).

More specifically, the technique proposed in [12] combines the concept of the generalized median with the batch computation of the SOM. The quadratic nature of the algorithm cannot be avoided, essentially because the utilized dissimilarity data are intrinsically described by a quadratic number of one-to-one similarities based on weighted Levenshtein distances. The self-organizing mixture model introduces an Expectation-Maximization (E-M) algorithm that yields topology preserving maps of data based on probabilistic mixture models, by assuming that there are several sources which generate the data. The model in [18] is such an approach that introduces a Bernoulli probabilistic SOM able to handle binary data by incorporating Hamming distances. An alternative approach so as to overcome the obstacle of introducing algebraic correlations between the symbols when handling symbolical sequences is orthogonal encoding [23]. This sparse encoding scheme has the disadvantage of increasing substantially the length of biological sequences (e.g. by a factor of 20 in the case of proteins). The merge SOM [24] devises a fusion of arbitrary lattice topologies with a noise-tolerant learning architecture which provides large flexibility and capacity due to the explicit representation of context. In this case, recursively computed Euclidean distances are used for realizing the learning algorithm. Furthermore, the approaches that are contained in the generalized taxonomy [2] and those that are described by the general framework for modeling recursive/recurrent SOMs [13] also depend on similar preprocessing stages for being in position to incorporate/analyze symbolical sequence data.

2. Self-organizing Hidden Markov Model map

2.1 Hidden Markov Model

In essence, a Hidden Markov Model (HMM) is a stochastic process generated by two interwoven probabilistic mechanisms, an underlying one (i.e. hidden) and an observable one (which produces the symbolical sequences). Thereinafter, we may denote each individual HMM as $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, where $\mathbf{A} = [a_{ij}]$, $\mathbf{B} = [b_j(k)]$ and $\boldsymbol{\pi}= [\pi_j]$ are the transition, emission and initial state probability stochastic matrices respectively.

Let $\{q_t\}_{t=1}^{\infty}$ be a homogeneous Markov chain, where each random variable q_t assumes a value in the state space $S = \{s_1, s_2, \ldots, s_N\}$. The conditional stationary probabilities are denoted as: $a_{ij} = P(q_t = s_j|q_{t-1} = s_i)$, $t>1$, $1\leq i \leq N$, $1\leq j \leq N$. Let $\{Y_t\}_{t=1}^{\infty}$ be a random process defined over the finite space $V = \{v_1, v_2, \ldots, v_M\}$ where, in general, $M \neq N$. The processes $\{q_t\}_{t=1}^{\infty}$ and $\{Y_t\}_{t=1}^{\infty}$ are related by the conditional probability distributions: $b_j(k) = P(Y_t = v_k|q_t = s_j)$, $t \geq 1$, $1\leq j \leq N$, $1\leq k \leq M$. In certain cases the emission probabilities' indexes are denoted as o_t and not as k. This interchange is made whenever the exact observation symbols are insignificant for the formulation under consideration, whereas, these values are considered to be given and specific. The initial state probability distribution is defined as: $\pi_j = P(q_1 = s_j)$, $1\leq j \leq N$.

2.2 The prototype

The SOHMMM introduces a hybrid integration of the SOM and the HMM. In essence, it implements a nonlinear structured mapping of sequence data onto the reference patterns of a low-dimensional array. In its basic form it produces a probability distribution graph of input data that summarizes the HMMs' distributions on the sequence space; subsequently, under certain conditions, the nonlinear statistical relationships between sequence data can be visually detected, investigated and interpreted.

Each reference pattern (neuron) e, $1 \leq e \leq E$ in the SOHMMM array consists of a HMM λ_e, also called reference HMM (E represents the overall number of neurons). Because the SOHMMM is conceptualized as a self-organizing methodology its algorithmic realization fuses the corresponding competition and cooperation processes. The leading partial process, viz. competition, finds the neuron with the best match (usually referred to as winner and indicated by the subscript c). Whereas, the following partial process, viz. cooperation, improves the match in the neighboring neurons located in the vicinity of the winner (Fig. 1) in an effort to gain some knowledge from the input sequence.

Initially, assume $O = o_1 o_2 \dots o_T$ is an input observation sequence. The two complementary processes of the SOHMMM algorithm are shown in Algorithm 1.

Algorithm 1 SOHMMM Learning Algorithm.

```
for $e = 1$ to $E$ do
    $\tilde{\alpha}_t^{(e)}(i) = \left[ \sum_{j=1}^{N} \hat{\alpha}_{t-1}^{(e)}(j) a_{ji}^{(e)} \right] b_i^{(e)}(o_t), 1 \leq i \leq N,\ 2 \leq t \leq T;$
    $sc_t^{(e)} = \left( \sum_{i=1}^{N} \tilde{\alpha}_t^{(e)}(i) \right)^{-1}, 1 \leq t \leq T;$
    $\hat{\alpha}_t^{(e)}(i) = sc_t^{(e)} \tilde{\alpha}_t^{(e)}(i), 1 \leq i \leq N,\ 2 \leq t \leq T;$
    $\tilde{\beta}_t^{(e)}(i) = \sum_{j=1}^{N} a_{ij}^{(e)} b_j^{(e)}(o_{t+1}) \hat{\beta}_{t+1}^{(e)}(j), 1 \leq i \leq N,\ t = T\text{-}1, \dots, 1;$
    $\hat{\beta}_t^{(e)}(i) = sc_t^{(e)} \tilde{\beta}_t^{(e)}(i), 1 \leq i \leq N,\ t = T\text{-}1, \dots, 1;$
end for
$c \leftarrow \arg\max_e \left\{ -\sum_{t=1}^{T} \log\left( sc_t^{(e)} \right) \right\};$
for $e = 1$ to $E$ do
    $w_{ij}^{(e)} \leftarrow w_{ij}^{(e)} + \eta(y) h_{ce}(y) \left\{ a_{ij} \sum_{l=1}^{T-1} \left[ \hat{\alpha}_l(i) \left( b_j(o_{l+1}) \hat{\beta}_{l+1}(j) - sc_l^{-1} \hat{\beta}_l(i) \right) \right] \Big|_{\lambda_e} \right\},$
        $1 \leq i \leq N,\ 1 \leq j \leq N;$
    $r_{jt}^{(e)} \leftarrow r_{jt}^{(e)} + \eta(y) h_{ce}(y) \left\{ \sum_{l=1}^{T} \left[ sc_l^{-1} \hat{\alpha}_l(j) \hat{\beta}_l(j) \left( I\left\{ o_l = t | \lambda \right\} - b_j(t) \right) \right] \Big|_{\lambda_e} \right\},$
        $1 \leq j \leq N,\ 1 \leq t \leq M;$
    $u_j^{(e)} \leftarrow u_j^{(e)} + \eta(y) h_{ce}(y) \left\{ \left[ \pi_j b_j(o_1) \hat{\beta}_1(j) - \pi_j \right] |_{\lambda_e} \right\}, 1 \leq j \leq N;$
    $a_{ij}^{(e)} \leftarrow e^{w_{ij}^{(e)}} \Big/ \sum_{l=1}^{N} e^{w_{il}^{(e)}}, 1 \leq i \leq N,\ 1 \leq j \leq N;$
    $b_j^{(e)}(t) \leftarrow e^{r_{jt}^{(e)}} \Big/ \sum_{l=1}^{M} e^{r_{jl}^{(e)}}, 1 \leq j \leq N,\ 1 \leq t \leq M;$
    $\pi_j^{(e)} \leftarrow e^{u_j^{(e)}} \Big/ \sum_{l=1}^{N} e^{u_l^{(e)}}, 1 \leq j \leq N;$
end for
```

The scaled forward and backward variables of reference HMM λ_e are denoted as $\hat{\alpha}_t^{(e)}(i)$ and $\hat{\beta}_t^{(e)}(i)$ respectively; whereas, $sc_t^{(e)}$are its scaling coefficients. Variable $y \geq 0$ is an integer, the discrete time coordinate. The function $\eta(y)$ plays the role of a scalar learning rate factor. The function $h_{ce}(y)$ acts as the neighborhood function; the width and form of $h_{ce}(y)$ define the stiffness of the elastic surface that is fitted to the input sequence data. $I\left\{ o_l = t | \lambda \right\}$ is the indicator function. The exponentially normalized variables of reference HMM λ_e are denoted as $w_{ij}^{(e)}$, $r_{jt}^{(e)}$ and $u_j^{(e)}$.

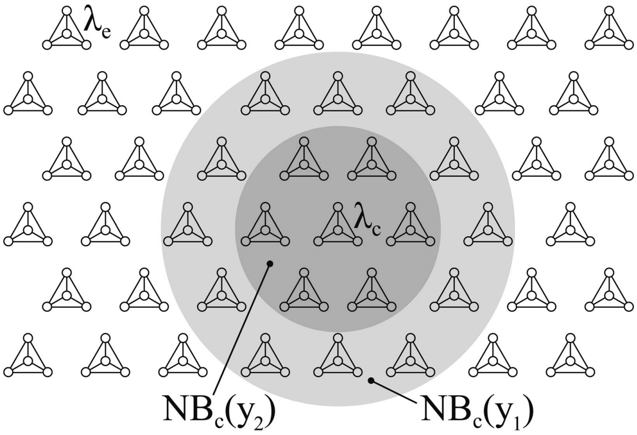

Fig. 1 *Paradigm of a hexagonal 8x6 SOHMMM lattice where each reference HMM λ_e is depicted as a fully-connected four-vertex six-edges graph. The circular shaded areas are an indicative winner neuron's (λ_c) closest topological neighborhoods (NB_c) at different points in time ($y_1 < y_2$). The wider light grey circular neighborhood corresponds to initial/earlier phases of the coarse learning procedure, whereas the narrower dark grey circular neighborhood corresponds to latter/final phases of the fine-tuning training phase.*

3. Visualization techniques and graphic displays

3.1 Sequence data density display

In essence, the SDDD depicts the number/amount of sequences which are assigned to each individual neuron of the SOHMMM mesh. Each sequence's winner (i.e. best matching) SOHMMM node is determined according to the maximum likelihood criterion. Furthermore, the corresponding graphic display, apart from the sequence data density, takes into consideration the type and dimensions of the SOHMMM lattice as well as the topology and positioning of the neurons. Each neuron is drawn with a size relative to the number of assigned sequences. An aftereffect of this procedure is that, in certain cases, clusters (of input sequence data) can be visually detected by searching for groups of topologically close/neighboring large size nodes separated by areas of small size (or zero size) nodes. It should be noted that a neuron is not drawn at all (usually referred to as zero size) if it is not the best match for any sequence. An abstract algorithmic formulation of the SDDD is given in Algorithm 2, where D is the overall/total number of available sequences, $O^{(d)} = o_1 o_2 \dots o_{T_d}$ is the d-th observation sequence which consists of T_d symbols. $P(O|\lambda)$ is the probability of sequence O conditioned on λ (likelihood). For technical reasons, probabilities can be very small. The solution is to work with the corresponding logarithms [15, 21].

Two remarks can be made by examining the SDDD algorithm. First, the resulting visualization depends on and takes into consideration the provided sequence data and the already trained SOHMMM. Second, one can defensibly claim that

Algorithm 2 Sequence Data Density Display.

```
for $d = 1$ to $D$ do
  for $e = 1$ to $E$ do
    $\tilde{\alpha}_t^{(e)}(i) = \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^{(e)}(j) a_{ji}^{(e)}\right] b_i^{(e)}(o_t), 1 \le i \le N,\ 2 \le t \le T;$
    $sc_t^{(e)} = \left(\sum_{i=1}^{N} \tilde{\alpha}_t^{(e)}(i)\right)^{-1}, 1 \le t \le T;$
    $\hat{\alpha}_t^{(e)}(i) = sc_t^{(e)} \tilde{\alpha}_t^{(e)}(i), 1 \le i \le N,\ 2 \le t \le T;$
    $\log P(O^{(d)}|\lambda_e) \leftarrow -\sum_{t=1}^{T} \log\left(sc_t^{(e)}\right);$
  end for
  $c \leftarrow \arg\max_e \left\{\log P(O^{(d)}|\lambda_e)\right\};$
  $numOfSeqs[c] \leftarrow numOfSeqs[c] + 1;$
end for
$maxNumOfSeqs \leftarrow \max_e \{numOfSeqs[e]\};$
$minNumOfSeqs \leftarrow \min_e \{numOfSeqs[e]\};$
for $e = 1$ to $E$ do
  $SDDD\_Paint(numOfSeqs[e], maxNumOfSeqs, minNumOfSeqs);$
end for
```

the SDDD is a straightforward unsupervised visualization technique since the only required/necessary information is the corpus of provided monomer sequences.

Experimental investigation/verification of the SDDD (and also, of the SLP) utilizes the globin protein family. Globins form a well-known family of heme-containing proteins that reversibly bind oxygen, and are involved in its storage and transport. The globin protein family is a large family which is composed of subfamilies. From crystallographic studies, all globins have similar overall three-dimensional structures but widely divergent sequences. The globin sequences used here were extracted from the iProClass protein knowledgebase [27], a database that provides extensive information integration of over 90 biological databases. In total, 560 proteins belonging to the three major globin subfamilies were retrieved (namely hemoglobin α-chains, hemoglobin β-chains, and myoglobins). The resulting globin data set's composition is 194 α-globins, 216 β-globins, and 150 myoglobins.

Initially, a hexagonal 6 × 7 mesh of 42 SOHMMM neurons was trained on the whole protein data set. The learning rate was defined as an exponentially decreasing function between 1.0 and 0.1, whereas the neighborhood kernels' standard deviations started with a value equal to half the largest dimension in the lattice and decreased linearly to one map unit. Also, the maximum total duration of both the ordering and tuning phases was set to ten epochs. The corresponding SDDD is illustrated in Fig. 2. An examination of the depicted display reveals that SDDD's two main goals are met satisfactorily. First, the number of each neuron's assigned sequences (viz. sequence data density) can be visually investigated. Second, three clusters are formed onto the SDDD, these consist of large size nodes' regions interrupted by small/zero size neurons' ravines and gaps. Furthermore,

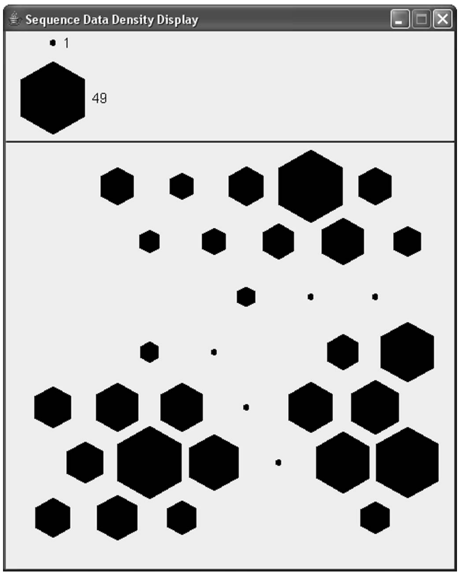

Fig. 2 *SDDD for a globin protein family which is based upon a* 6 × 7 *SOHMMM array. The upper part contains a legend that illustrates the utilized analogy between the size of a hexagon and the corresponding number of sequences (i.e. the smallest in size hexagon describes only one globin, whereas the largest size hexagon forty-nine). In the lower part the actual SDDD can be examined. Each hexagon, regardless of size, represents a SOHMMM neuron. The missing hexagons actually represent neurons that do not describe any sequence (zero size). The more sequences are assigned to a specific neuron, the largest is the size of its hexagon. It is evident that there are concentrations of neurons with an increased number of assigned sequences and zones of neurons that contain a small number of, or even no, sequences. The former represent the detected clusters (viz. the three globin subfamilies in this case), whereas the latter comprise the boundaries between these clusters.*

one can easily verify that these three clusters correspond to each individual globin subfamily. Protein class information that is excluded from the training and visualizing procedures can be used to perform a posterior identification/labeling of each SOHMMM node. The outcome of this procedure can be found in Fig. 3, the one-to-one correspondence between a depicted cluster and a globin subfamily is evident.

The Java language has been utilized for programming both the SDDD and SLP visualization interfaces. The main reason for this choice was the built-in platform independent support of graphics. The majority of supplied figures are screenshots of the implemented graphical user interface. The SDDD's interface (apart from the top-left legend) provides additional features and functionality. The number of assigned sequences is displayed by moving the mouse pointer over a SOHMMM node, whereas by left-clicking a neuron the user has the options of viewing or saving: (1) the assigned sequences' amino acid or nucleotide descriptions; (2) the assigned

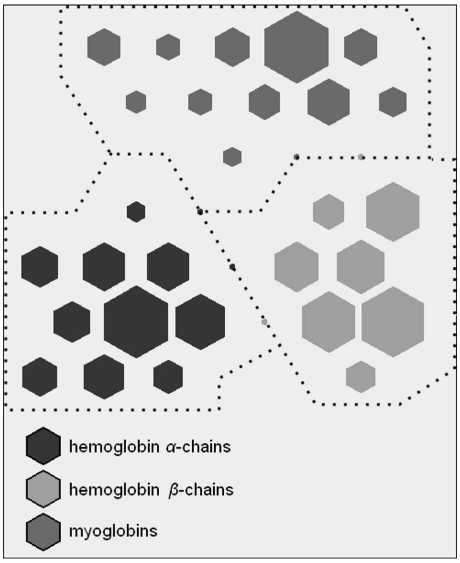

Fig. 3 *The devised SDDD visualization (i.e. the lower part of Fig. 2) enriched with information obtained from posterior labeling of depicted hexagons. This post-processing approach acts as a proof-of-concept so as to demonstrate that the SDDD's identified clusters have a one-to-one correspondence with the globin subfamilies which are existent in the data set. The dotted lines mark the boundaries of the α-globin, β-globin, and myoglobin detected clusters.*

sequences' identifiers; (3) the corresponding HMM's parameters. This information can potentially be used for studying the profile and statistical properties of the assigned sequences [1, 6, 17, 20, 22].

An intrinsic drawback of the SDDD originates from its dependence from input sequence data. In certain cases, where sequences that belong to specific clusters are orders of magnitude larger in numbers compared to the remaining clusters' sequences, visual detection of the latter clusters is obstructed (since these are suppressed on the display). Appropriate thresholds could be employed for overcoming this difficulty but their determination would deviate considerably from the unsupervised framework, and thus, are avoided.

3.2 Sequence likelihood projection

Practically, the SLP is a graphic representation of the likelihood magnitudes (of a given sequence) with respect to each individual SOHMMM neuron. As expected, the corresponding visualization incorporates the type and dimensions of the SOHMMM array as well as the topology and positioning of the respective nodes. More specifically, each neuron's coloring is analogous to the likelihood value it yields; the color itself is selected relatively to the employed color scale. A potential aftereffect of this process is that a cluster might be visually traced (given the examined sequence belongs to a cluster which is represented) by searching for groups of topologically close/neighboring high likelihood nodes separated by areas of low likelihood nodes. A generic formulation of the SLP algorithm is shown in Algorithm 3, where $OS = \left\{O^{(1)}, O^{(2)}, \ldots, O^{(D)}\right\}$ is the set of D available observation sequences.

Algorithm 3 Sequence Likelihood Projection.

```
$O \leftarrow$ sequenceSelection($OS$);
for $e = 1$ to $E$ do
    $\tilde{\alpha}_t^{(e)}(i) = \left[\sum_{j=1}^{N} \hat{\alpha}_{t-1}^{(e)}(j) a_{ji}^{(e)}\right] b_i^{(e)}(o_t), 1 \le i \le N,\ 2 \le t \le T;$
    $sc_t^{(e)} = \left(\sum_{i=1}^{N} \tilde{\alpha}_t^{(e)}(i)\right)^{-1}, 1 \le t \le T;$
    $\hat{\alpha}_t^{(e)}(i) = sc_t^{(e)} \tilde{\alpha}_t^{(e)}(i), 1 \le i \le N,\ 2 \le t \le T;$
    $\log P(O|\lambda_e) \leftarrow -\sum_{t=1}^{T} \log\left(sc_t^{(e)}\right);$
    $likelihood[e] \leftarrow \log P(O|\lambda_e);$
end for
$maxLikelihood \leftarrow \max_e \{likelihood[e]\};$
$minLikelihood \leftarrow \min_e \{likelihood[e]\};$
for $e = 1$ to $E$ do
    SLP_Paint($likelihood[e], maxLikelihood, minLikelihood$);
end for
```

In this case, also, two comments can be made. The resulting graphic display is based upon and incorporates the selected sequence and the already trained SOHMMM. Moreover, one can claim that the SLP is a direct unsupervised projection technique since the only required/necessary information is the provided monomer sequence.

The present experimental examination of the SLP was conducted by using the same SOHMMM that had been trained previously on the globin protein family. The resulting SLP, for a sequence belonging to one of the three subfamilies (hemoglobin α-chains), is illustrated in Fig. 4. The initially defined objectives are (at a certain extent) achieved. The sequence's likelihood value distribution (or likelihood landscape) is represented visually. More important perhaps, one cluster is formed onto the SLP. This cluster consists of cohesive areas of high likelihood neurons surrounded by regions of low likelihood neurons. Furthermore, by keeping in mind the fact that this SLP describes a hemoglobin α-sequence; there is an evident analogy/correspondence with the detected α-globin cluster of the SDDD visualization (Fig. 2).

Per contra, a cytoglobin (Xenopus laevis) not used during the adaptation procedure of the SOHMMM, viz. not represented explicitly on the SOHMMM plane, is depicted in Fig. 5. In this case, instead of observing cohesive regions with well-defined boundaries, we see that sparse mosaic-like parcellations emerge. SLP's resulting visualization appears unstructured and unformed. This is justifiable since the specific SOHMMM has not been trained for modeling the cytoglobin subfamily

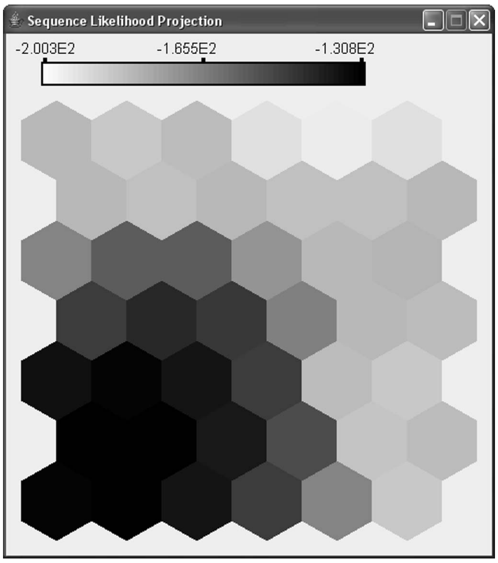

Fig. 4 *Characteristic SLP, which is based upon a 6 × 7 SOHMMM array, that corresponds to an α-globin. In the upper part the displayed color scale is in analogy to the log-likelihood values (i.e. higher values correspond to darker shades and vice versa). Each hexagon, regardless of color, represents a SOHMMM neuron. In the lower part the actual SLP for the α-globin case is depicted. A region rich in darker hexagons (high likelihood neurons) surrounded by hexagons of lighter shades (low likelihood neurons) is easily detectable. These high likelihood neurons reside within the boundaries of the α-globin's container cluster. Furthermore, the low likelihood neurons are exclusively only those that belong either to the boundaries of the cluster under consideration or to the remaining clusters that do not represent the α-globin subfamily.*

or for clustering the respective chain molecules. Nevertheless, even in this case, the highest likelihood values are located in areas which do not contain any of the three known clusters (Fig. 2). We believe that this is an additional indirect proof of the fact that the SOHMMM produces a nonlinear, ordered mapping of sequence data. Even though, the cytoglobin is applied for the first time it is not assigned falsely to a HMM neuron corresponding to one of the three previously identified clusters, instead the SOHMMM node that best describes the cytoglobin sequence does not represent any α-globin, β-globin or myoglobin cluster.

As mentioned previously the SLP visualization interface has been programmed in Java. Likewise, all figures are screenshots of the implemented graphical user interface. The SLP's interface includes a color scale legend (which is in analogy to the minimum and maximum log-likelihood values) and has functions for retrieving the examined sequence's amino acid/nucleotide chain as well as its identifier. Also, the exact likelihood magnitude/value is displayed when positioning the mouse pointer over each SOHMMM node. Last, by left-clicking a neuron the user has the option to view or save the respective HMM's coefficients/parameters.

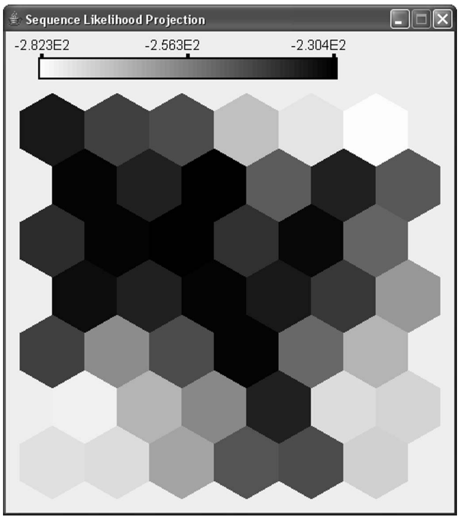

Fig. 5 *Indicative outcome of a SLP, based upon the same* 6 × 7 *SOHMMM, corresponding to a cytoglobin chain molecule. The specific cytoglobin is actually an outlier for the clustering problem under consideration since it does not belong to any of the subfamilies modeled by the algorithm. Nevertheless, along a pure unsupervised procedure, the SLP demonstrates generalization ability by not clustering the cytoglobin as belonging to one of the previously detected clusters (Fig. 2), but instead produces a likelihood tessellation which is indicative of the cases where the investigated sequence does not belong to the modeled sequence distribution.*

4. Application scenario

Splice junctions mark transitions from expressed to unexpressed gene regions and vice versa, and they are thus important for the assembly of structural and regulatory proteins that constitute and control biochemical metabolisms. Splice site recognition is therefore a topic of interest for the understanding of genotype/phenotype relationships. The UCI machine learning repository [19] contains a data set for primate (eukaryotic) splice junction determination.

In an effort to put the SDDD's and the SLP's characteristics and capabilities in context we employ the devised unsupervised learning algorithm for training a hexagonal 9 × 4 SOHMMM array; all the remaining parameters are identical to those used previously. The exact problem posed in this practical application is to recognize the exon-intron boundaries (frequently called donor splice sites) and the intron-exon boundaries (acceptor splice sites). The sequence data set's composition is 767 donors and 768 acceptors. The produced visualizations are shown in Figs. 6–8.

Initially, one can observe that the SDDD (in synergy with the underlying SOHMMM) identifies the existence of the two splice junction clusters (namely the acceptor cluster and the donor cluster). Moreover, alongside a strict unsupervised learning procedure (no prior knowledge nor category information are used in any stage), the recognized clusters are directly traced/identified visually (Fig. 6).

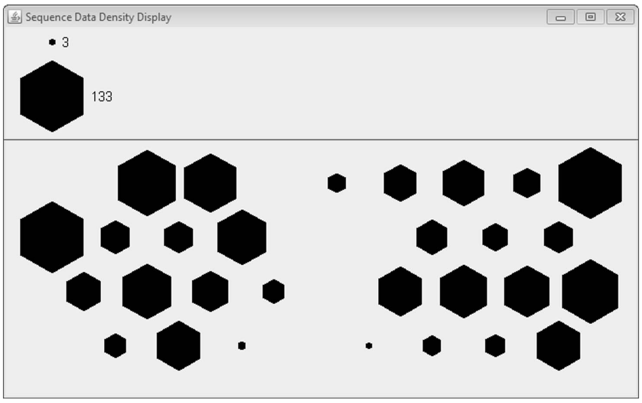

Fig. 6 *SDDD of a 9×4 SOHMMMM lattice for the splice junction recognition problem. Two clusters have been detected that correspond to the exon-intron boundaries (donor splice sites) and the intron-exon boundaries (acceptor splice sites). The donor splice sites' cluster is located on the right of the SDDD, whereas the acceptor splice sites' cluster is on the left.*

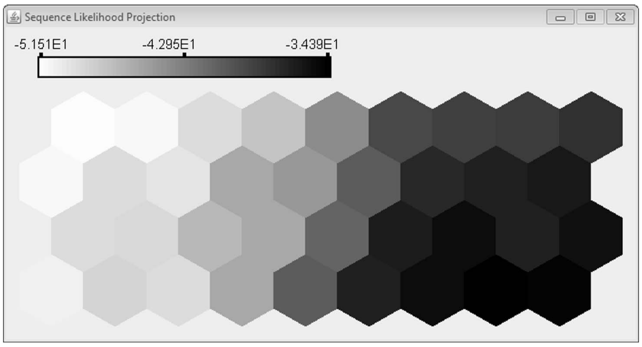

Fig. 7 *Representative SLP of a sequence containing an exon-intron boundary.*

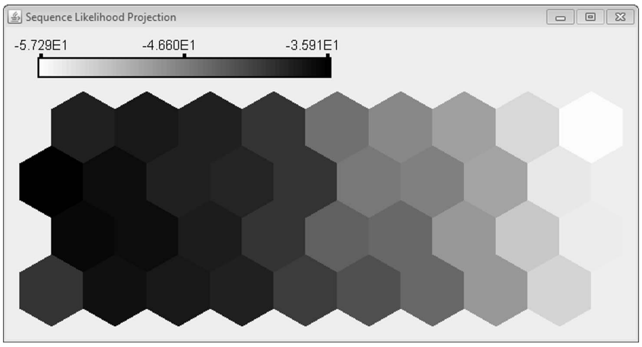

Fig. 8 *Paradigm of a SLP visualization for an acceptor splice site sequence.*

The interpretation of the SLPs' results (Figs. 7 and 8) is dual. They can either be considered as assignments of unknown sequences to the previously detected clusters (viz. sequences with exon-intron boundaries are assigned to the donor cluster and sequences with intron-exon boundaries are assigned to the acceptor cluster), or they can be treated as an additional verification/proof of the discovered clusters (since each one of the 1535 sequences produces high likelihoods in exactly one out of the two distinct areas of the SOHMMM mapping, and these two regions coincide with the recognized clusters).

Subsequently, under the assumption that class information is given a posterior labeling of the traced clusters can be performed on the SDDD (as in Fig. 3). In such a case, an unknown or orphan splice junction is classified as belonging to the category designated by the SOHMMM neuron yielding the highest likelihood, and (as shown previously) is clustered accordingly. It is interesting that the SLP's visualization goes beyond the mapping of the highest likelihood SOHMMM node by depicting the overall likelihood landscape (of the examined/analyzed sequence), thus, resulting in a richer and comprehensive description.

5. Conclusion

The present study has developed mapping techniques and graphic displays to demonstrate and visualize sequence data clustering results. Epigrammatically, the SDDD estimates and subsequently devises a sequence data density graphic representation, whereas the SLP produces a sequence likelihood landscape. The SDDD and SLP projections represent the input sequence data in a lower-dimensional space in such a way that the clusters and statistical relations of data elements are preserved as faithfully as possible, thus, making the SOHMMM's capabilities accessible to a wider range of interdisciplinary tasks. It has been demonstrated that the SDDD and the SLP in synergy with the SOHMMM integrate cluster visualization and nonlinear mapping (on a low-dimensional lattice) in a unified functional framework; thus making the produced results visually accessible, verifiable and interpretable. The proposed techniques' experimental testing and verification has been performed both on amino acid and nucleotide sequences.

Acknowledgement

The authors would like to thank graphic artist and biologist Vanessa Ferle for her substantial assistance and fundamental contribution in all graphic design stages.

References

[1] BALDI P., BRUNAK S. *Bioinformatics: the machine learning approach.* MIT press, 2001, ISBN 026202506X.

[2] BARRETO G.D.A., ARAÚJO A.F., KREMER S.C. A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case. *Neural Computation.* 2003, 15(6), pp. 1255−1320, doi: 10.1162/089976603321780281.

[3] BOGDAN M., ROSENSTIEL W. Detection of cluster in Self-Organizing Maps for controlling a prostheses using nerve signals. In: *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'2001)*, Bruges, Belgium. D-Facto public., 2001, pp. 131–136.

[4] BRUGGER D., BOGDAN M., ROSENSTIEL W. Automatic cluster detection in Kohonen's SOM. *IEEE Transactions on Neural Networks*. 2008, 19(3), pp. 442–459, doi: 10.1109/TNN.2007.909556.

[5] DU K.-L. Clustering: A neural network approach. *Neural Networks*. 2010, 23(1), pp. 89–107, doi: 10.1016/j.neunet.2009.08.007.

[6] DURBIN R., EDDY S.R., KROGH A., MITCHISON G. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998. ISBN 113945739X.

[7] FERLES C., SIOLAS G., STAFYLOPATIS A. Scaled on-line unsupervised learning algorithm for a SOM-HMM hybrid. In: E. Gelenbe, R. Lent, G. Sakellari, eds. *Computer and Information Sciences II*. London: Springer, 2011, pp. 533–537, doi: 10.1007/978-1-4471-2155-8_68. Proceedings of the 26th International Symposium on Computer and Information Sciences.

[8] FERLES C., SIOLAS G., STAFYLOPATIS A. Scaled self-organizing map – hidden Markov model architecture for biological sequence clustering. *Applied Artificial Intelligence*. 2013, 27(6), pp. 461–495, doi: 10.1080/08839514.2013.805598.

[9] FERLES C., STAFYLOPATIS A. A hybrid self-organizing model for sequence analysis. In: *2008 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'08)*, Dayton, Ohio. IEEE, 2008, vol. 2, pp. 105–112, doi: 10.1109/ICTAI.2008.108.

[10] FERLES C., STAFYLOPATIS A. Sequence clustering with the Self-Organizing Hidden Markov Model Map. In: 8th IEEE International Conference on BioInformatics and Bio-Engineering (BIBE 2008), Athens, Greece. IEEE, 2008, pp. 1–7, doi: 10.1109/BIBE.2008.4696720.

[11] FERLES C., STAFYLOPATIS A. Self-organizing hidden markov model map (SOHMMM). *Neural Networks*. 2013, 48, pp. 133–147, doi: 10.1016/j.neunet.2013.07.011.

[12] HAMMER B., HASENFUSS A. Topographic mapping of large dissimilarity data sets. *Neural Computation*. 2010, 22(9), pp. 2229–2284, doi: 10.1162/neco_a_00012.

[13] HAMMER B., MICHELI A., SPERDUTI A., STRICKERT M. A general framework for unsupervised processing of structured data. Neurocomputing. 2004, 57, pp. 3–35, doi: 10.1016/j.neucom.2004.01.008.

[14] KOHONEN T. *Self-organizing maps*. Berlin: Springer Verlag, 2001. Vol. 30 of Springer Series in Information Sciences. ISBN 3540679219.

[15] KOSKI T. *Hidden Markov models for bioinformatics*. Dordrecht: Kluwer Academic Publishers, vol. 2, 2001. ISBN 1402001363.

[16] KRAAIJVELD M.A., MAO J., JAIN A.K. A nonlinear projection method based on Kohonen's topology preserving maps. *IEEE Transactions on Neural Networks*. 1995, 6(3), pp. 548–559, doi: 10.1109/72.377962.

[17] KROGH A., BROWN M., MIAN I.S., SJÖLANDER K., HAUSSLER D. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*. 1994, 235(5), pp. 1501–1531, doi: 10.1006/jmbi.1994.1104.

[18] LEBBAH M., ROGOVSCHI N., BENNANI Y. BeSOM: Bernoulli on self-organizing map. In: *2007 International Joint Conference on Neural Networks (IJCNN 2007)*, Orlando, Florida. IEEE, 2007, pp. 631–636, doi: 10.1109/IJCNN.2007.4371030.

[19] LICHMAN M. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. 2013 [accessed 2016-05-05]. Available from: http://archive.ics.uci.edu/ml/index.html

[20] MOUNT D.W. *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor, New York: Cold Spring Harbor Laboratory Press, 2004. ISBN 0879697121.

[21] RABINER L.R. A tutorial on hidden Markov models and selected applications in speech recognition. In: *Proceedings of the IEEE*. 1989, 77(2), pp. 257–286, doi: 10.1109/5.18626.

[22] SHARMA K. *Bioinformatics: Sequence alignment and Markov models*. McGraw Hill Professional, 2008. ISBN 0071593071.

[23] SOMERVUO P.J. Online algorithm for the self-organizing map of symbol strings. *Neural Networks*. 2004, 17(8), pp. 1231–1239, doi: 10.1016/j.neunet.2004.08.004.

[24] STRICKERT M., HAMMER B. Self-organizing context learning [online]. In: *Proceedings - European Symposium on Artificial Neural Networks (ESANN'2004)*, Bruges, Belgium. D-side public., 2004, pp. 39–44 [viewed 2016-06-02]. Available from: https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2004-73.pdf

[25] TAŞDEMIR K. Graph based representations of density distribution and distances for self-organizing maps. *IEEE Transactions on Neural Networks*. 2010, 21(3), pp. 520–526, doi: 10.1109/tnn.2010.2040200.

[26] ULTSCH A. Maps for the visualization of high-dimensional data spaces [online]. In: *Proceddings of the workshop on Self organizing Maps*. 2003, pp. 225–230 [viewed 2016-06-02]. Available from: https://www.researchgate.net/publication/228706090_Maps_for_the_visualization_of_high-dimensional_data_spaces

[27] WU C.H., HUANG H., NIKOLSKAYA A., HU Z., BARKER W.C. The iProClass integrated database for protein functional analysis. *Computational biology and chemistry*. 2004, 28(1), pp. 87–96, doi: 10.1016/j.compbiolchem.2003.10.003.

[28] XU R., WUNSCH D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*. 2005, 16(3), pp. 645–678, doi: 10.1109/tnn.2005.845141.

[29] YIN H. ViSOM-a novel method for multivariate data projection and structure visualization. *IEEE Transactions on Neural Networks*. 2002, 13(1), pp. 237–243, doi: 10.1109/72.977314.