Bioinformatics, 31(5), 2015, 736–744
doi: 10.1093/bioinformatics/btu708
Advance Access Publication Date: 24 October 2014
Original Paper

OXFORD

Data and text mining

A novel representation of genomic sequences for taxonomic clustering and visualization by means of self-organizing maps

Soledad Delgado[1,*], Federico Morán[2], Antonio Mora[3,4], Juan Julián Merelo[3,4] and Carlos Briones[5,6]

[1]Department of Information Structure and Organization, Universidad Politécnica (UPM), Madrid 28031, [2]Department of Biochemistry and Molecular Biology I, Universidad Complutense (UCM), Madrid 28040, [3]Department of Computer Architecture and Computer Technology, Universidad de Granada (UGR), Granada 18071, Spain, [4]CITIC, Campanillas, Malaga 29590, Spain, [5]Department of Molecular Evolution, Centro de Astrobiología (CSIC-INTA), Torrejón de Ardoz, Madrid 28850 and [6]Centro de Investigación Biomédica en Red de enfermedades hepáticas y digestivas (CIBERehd), Barcelona 08036, Spain

*To whom correspondence should be addressed.
Associate Editor: Jonathan Wren

Received on July 17, 2014; revised on September 30, 2014; accepted on October 21, 2014

Abstract

Motivation: Self-organizing maps (SOMs) are readily available bioinformatics methods for clustering and visualizing high-dimensional data, provided that such biological information is previously transformed to fixed-size, metric-based vectors. To increase the usefulness of SOM-based approaches for the analysis of genomic sequence data, novel representation methods are required that automatically and bijectively transform aligned nucleotide sequences into numeric vectors, dealing with both nucleotide ambiguity and gaps derived from sequence alignment.

Results: Six different codification variants based on Euclidean space, just like SOM processing, have been tested using two SOM models: the classical Kohonen's SOM and growing cell structures. They have been applied to two different sets of sequences: 32 sequences of small sub-unit ribosomal RNA from organisms belonging to the three domains of life, and 44 sequences of the reverse transcriptase region of the *pol* gene of human immunodeficiency virus type 1 belonging to different groups and sub-types. Our results show that the most important factor affecting the accuracy of sequence clustering is the assignment of an extra weight to the presence of alignment-derived gaps. Although each of the codification variants shows a different level of taxonomic consistency, the results are in agreement with sequence-based phylogenetic reconstructions and anticipate a broad applicability of this codification method.

Contact: sole@eui.upm.es

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Genomic sequence data is continuously produced and processed in fields ranging from biomedicine to biotechnology. Over the last decade, the completion of several genome sequence projects and the advent of next generation, high-throughput sequencing techniques have dramatically increased the amount of sequence data stored in databases. As a consequence, fast algorithms and methods are being developed for converting the growing number of nucleotide sequences into useful information, such as data classes and clusters, sets of the most relevant features or hidden relationships among data, to cite the most common ones. Within this framework, the need for automatic massive sequence classification techniques is becoming more acute, requiring methods that are able to efficiently process and classify genomic data.

One commonly used technique for data clustering and visualization is Kohonen's self-organizing map (SOM; Kohonen, 2001), which allows processing a variety of biological data (Boyle *et al.*, 2014; Chavez-Alvarez *et al.*, 2014). SOM performs an unsupervised clustering process, i.e. during the training the patterns are spatially organized considering only their homology, without knowing the class to which they belong (Vesanto and Alhoniemi, 2000; Xu and Wunsch, 2005; Astel *et al.*, 2007). Therefore, the use of an unsupervised method like SOM seems very appropriate for genomic sequence clustering and, thus, representation, since it is able to conduct exploratory data analysis and visualization without the need for calibration or classification of the information to be processed. In comparison with other clustering methods, SOM adds the feature of spatially ordering the sequences on a two-dimensional (2D) map, thus, providing not only the clustering of the sequences into groups, but also the similarity or dissimilarity with all other groups of sequences. This could be very useful for organizing large sequence databases, as well as for quickly and accurately ordering newly obtained sequences.

SOM creates a model composed of neurons, each with a prototype vector. The neurons are organized into a grid, which optimally represents the set of input data included into the training set (Murtagh and Hernandez-Pajares, 1995). This model is self-organized by a training algorithm based on two fundamental concepts: the neighborhood relationship among the prototype vectors placed in such a grid, and the smooth adaptation of prototype vectors over the vectors in the training set. Traditional SOM works with continuous numeric input space and Euclidean distances, and it cannot be directly used for processing strings of symbols such as nucleotide or amino acid sequences. Different authors, including Kohonen and Somervuo (2002) have highlighted this fact as a major limitation of the method. Two alternative approaches have been suggested in the literature to solve this problem: to transform the training algorithm to be able to work with a discrete input space, and to convert the character sequences into numeric vectors and then use a traditional SOM.

The String SOM model (Kohonen and Somervuo, 2002) is comprised within the first option. It relies on the FASTA method (Pearson and Lipman, 1988) to compute similarity between sequences, and either the generalized median or the set median is used to adapt the string prototype vectors. However, this model presents several limitations. On the one hand, the algorithm for calculating the generalized median, i.e. the sequence (over the set of all possible sequences) that minimizes the sum of distances to every sequence of the set, is extremely time-consuming (Jiang *et al.*, 2004; Solnon and Jolion, 2007). On the other hand, the set median, where string prototype values are restricted to the set of training sequences, constitutes a severe restriction, in particular for small data sets, because it can infer non-representative prototype strings. Furthermore, String SOM is highly sensitive to prototype initialization (Kohonen and Somervuo, 2002), forcing the prototype vectors to be roughly ordered in advance.

Transforming sequences into numeric vectors and training a traditional SOM is the second option. The conversion of sequences into same-length vectors in a meaningful way is a computational challenge, and two types of approaches are possible to accomplish this goal. The first type transforms sequences of variable length into fixed-length numeric vectors without a preliminary alignment (Dozono, 2014; Almeida and Vinga, 2009; Fankhauser and Mäser, 2005). However, most of the sequence-vector pairs are not bijective, i.e. the position of the nucleotide in the original (DNA or RNA) sequence is not embedded in the final vector, since it often represents either a frequency histogram or a picture. Thus, the information residing in the succession of elements in the sequence of characters is lost and it is not possible to fully reconstruct the original sequence from the numeric vector sequence. This can constitute a constraint when SOM method is used to process these numeric vectors, because in traditional SOM, based on a continuous numeric space, prototype vectors virtually never match any of the training vectors. SOM prototype and training vectors share the same nature and dimension, and conceptually each prototype vector symbolizes the centroid of the training vectors that are within its influence region. Therefore, when a SOM is trained using nucleotide sequences previously transformed into numeric vectors, a bijective transformation algorithm becomes desirable to analyse the prototype vectors of the SOM in their nucleotide sequence format. The second type of methods for transforming sequences into numeric vectors requires a preliminary alignment of the sequences. They are usually bijective, but typically produce very long numeric vectors (Andrade *et al.*, 1997; Nantasenamat *et al.*, 2009; Afreixo *et al.*, 2009). Kwan and Arniker (2009) presented an analysis of some bijective numerical representation methods of DNA sequences, highlighting the final length of the numerical vectors. The combination of the bijective DNA transformation methods with traditional SOMs is reinforced by the following facts: (i) SOM prototype vectors are not limited to the set of training vectors; (ii) the training algorithm is computed efficiently since it works with numeric vectors instead of strings; (iii) given its bijective nature, it can alternatively work with nucleotide sequences or numeric vectors.

The work reported here presents a new bijective method with six different variants for coding nucleotide sequences. These have been tested using two numeric SOM models: the classical Kohonen SOM (Kohonen, 2001), as well as an improved version of growing cell structures (GCS; Fritzke, 1994; Delgado *et al.*, 2011). The former is known for its usefulness in graphical exploratory data analysis, and the latter for providing better clustering results due to its flexible architecture. Two sets of DNA sequences have been used to validate the coding variants of the method: 32 small sub-unit ribosomal RNA (SSU rRNA) from organisms belonging to the three domains of life, and 44 sequences of the reverse transcriptase region of the *pol* gene of human immunodeficiency virus type 1 (HIV-1) belonging to different groups and sub-types. The six variants of the proposed transforming method have been compared. We found that the two SOM models distribute the encoded sequences according to their taxonomic diversity and phylogenetic relationships. In particular, two of the six coding variants have produced highly accurate results.

2 Materials and Methods

2.1 Sequence codification

A DNA sequence can be viewed as a string of symbols of a finite alphabet {A, G, C, T}: the nucleotides adenine, guanine, cytosine and thymine (with uracil—U—instead of T in the case of RNA). When more than one kind of nucleotide can appear at a given position in the string, as the consequence of either a technical problem (e.g., an ambiguous nucleotide assignment during the sequencing process), or the inherent variability within an heterogeneous population (e.g. a mixture of nucleotides at a certain position of the consensus sequence of an RNA virus population), additional symbols must be used, following IUPAC nomenclature rules: R (A or G), Y (C or T), K (G or T), M (A or C), S (G or C), W (A or T), B (G, C or T), D (A, G or T), H (A, C, or T), V (A, G or C), N (A, G, C, or T). The bijective transformation method presented in this article requires a preliminary alignment of the sequences to be compared, so it is necessary to consider an additional symbol for the alphabet, the 'gap' or '-'. The alignment of the sequences yields vectors of the same length, since this process fills with gaps the extra positions that are present only in a subset of the aligned sequences. In summary, the DNA alphabet for any aligned sequence consists of 16 symbols: {A, G, C, T, R, Y, K, M, S, W, B, D, H, V, N, -}.

The codification method used here is related to that proposed by Lo *et al.* (2007), consisting of mapping the symbols A, G, C and T on the four vertices of a regular tetrahedron (Supplementary Fig S1A). Each nucleotide is then transformed into a numeric vector, representing the 3D Euclidean coordinates of its corresponding vertex. Here we propose an extended DNA codification method including the eleven nucleotide ambiguity symbols. They are coded using the 3D coordinates corresponding to the midpoint of the edge joining the vertices occupied by the two possible nucleotides (for R, Y, K, M, S and W symbols), the centroid of the plane defined by the vertices that code the three possible nucleotides (for B, D, H and V symbols), or the centroid of the tetrahedron for the N symbol (Supplementary Fig S1A).

Additionally, two novel variables have been implemented to differently weight each of the possible nucleotide mutations in a DNA sequence: (a) tetrahedron symmetry, affecting any nucleotide substitution, and (b) gap codification (symbol '-'), related to the insertion/deletion mutations, also called 'indels'. Regarding the first factor, two further options are proposed: (a1) a regular tetrahedron with distance 1 between all its vertices, or (a2) an irregular tetrahedron with distance 1 for the edges joining the vertices A–G (purine nucleotides) and C–T (pyrimidine nucleotides), and distance 2 for those joining A–C and G–T (Supplementary Fig S1). Thus, in the regular tetrahedron all kinds of nucleotide mutations have the same weight. In turn, using the irregular one, a transition mutation (that produced between either the purine nucleotides or the pyrimidine ones) is considered closer than a transversion (that converting a purine into a pyrimidine or vice versa), since the latter is less thermodynamically favored and consequently appears less frequently in nature. Indeed, most of the computer programs for inferring molecular phylogenies (e.g. the PHYLIP package; http://evolution.genetics.washington.edu/phylip.html) include the Kimura 2-parameter correction method for estimating distances between sequences, which weights the transitions/transversions ratio as 2 (Kimura, 1980).

With respect to gap coding, two options are also considered: (b1) each gap is computed as a substitution mutation, assigning it the 3D coordinates of the centroid of the tetrahedron (equivalent to the ambiguous symbol N), or (b2) the gap reflects a new class of genetic rearrangement (one 'indel' in at least one of the aligned sequences) that requires weighting it in a different way. To encode the latter option, a fourth dimension (4D) has been incorporated to the 3D numeric vector, taking the value 0 for all the alphabet symbols except for the gap, whose three first coordinates remain those of the centroid of the tetrahedron. Two possible values have been considered for the fourth dimension of the gap: (b2.1) near (at a distance 2), and (b2.2) far (at a distance 4). The combination of the tetrahedron symmetry and gap coding options has led to the six DNA coding variants presented in this article: *3DReg*, *4DRegNear*, *4DRegFar*, *3DIrreg*, *4DIrrNear* and *4DIrrFar*. The coordinates corresponding to each symbol in these coding variants are shown in Supplementary Table S1.

Given an aligned DNA sequence of L nucleotides, the use of any of the *3D* coding variants results in a numeric vector of length $3 \times L$, and the use of any of the 4D ones produces a $4 \times L$-long numeric vector. The original DNA sequence can be retrieved from the numeric vector by transforming each subsequence of j numbers (where j takes the value 3 for 3D codifications and 4 for 4D codifications) into an alphabet symbol, calculating the closest coding position (nucleotide, ambiguous nucleotide or gap) in the tetrahedron by means of Euclidean distance (Supplementary Fig S2 shows an example illustrating the transformation of a DNA sequence into the *3DReg* codified vector, and vice versa, the transformation of a *3DReg* codified vector into the corresponding DNA sequence).

2.2 Datasets

To make a comparative study of the six coding variants proposed in this work, two sets of nucleotide sequence data have been used. The first of them (termed 'Biodiversity dataset') includes sequences of SSU rRNA (the molecule considered as the best 'molecular clock' for phylogenetic studies since the pioneering work of Woese and Fox (1977; Sapp, 2009), of 32 organisms belonging to the three major groups or domains of cellular life: Archaea (18 sequences), Bacteria (8) and Eukarya (6). This set of data has been previously used to compare the phylogenetic reconstructions based on sequence data with those derived from functional information (Briones *et al.*, 2005). Interestingly, the group of organisms included in this biodiversity set shows a different degree of taxonomic diversity within each domain, thus, allowing an evaluation of the limit of resolution of the different codification variants of the clustering method used here. The Biodiversity dataset includes a large number of gaps derived from the sequence alignment (see Supplementary Material; Supplementary Table S2 for details).

The second set (termed 'HIV dataset') contains 44 complete sequences of the RT region of the *pol* gene of HIV-1 belonging to the three phylogenetic groups of this virus (M, O and N), and to all the subtypes within the group M (A, B, C, D, F, G, H, J, K). Being a virus with an RNA genome, HIV-1 displays a very high mutation rate and the genomic sequences of different isolates show a large variability (Domingo *et al.*, 2012). However, RT-coding sequence of the HIV-1 genome is one of the most conserved regions, thus being a very appropriate genetic marker for comparing taxonomically distant viral isolates. The HIV dataset includes a low number of alignment-derived gaps (see Supplementary Material; Supplementary Table S3 for details).

The whole nucleotide sequences of the Biodiversity and HIV datasets are shown in Supplementary Table S4. As a reference classification method for the sequences included in both datasets, sequence-based phylogenetic analyses have been performed (see Supplementary Material for details).

3 Results and Discussion

3.1 Reference classification by phylogenetic methods

The Biodiversity dataset shows three clusters of sequences that are clearly distinguished, statistically supported by bootstrap values higher than 80% (see Supplementary Fig S3). They correspond to organisms belonging to the domains Archaea, Bacteria and Eukarya. Within Archaea, the phyla *Crenarchaeota* and *Euryarchaeota* (see Supplementary Table S2) are distinguished, and classes within them are also shown. Although the number of sequences within Bacteria is smaller, the phyla *Proteobacteria* and *Cyanobacteria* are also clearly separated, with the sub-division in classes reflected by the branching order of the clustered species. The six eukaryotic sequences included in this dataset are also clustered accordingly to their taxonomic relatedness, with the sequences belonging to the kingdoms *Fungi* and *Viridiplantae* grouping together.

In HIV dataset, sequences belonging to the group O are separated from those of the groups N and M (Supplementary Fig S4). Within group M, all the sub-types (Supplementary Table S3) are clearly distinguished, with the pairs F–K and B–D grouping together. The clustering shows sub-subtype resolution, since the separation between sub-subtypes A1–A2 and F1–F2 is supported by high bootstrap values within sub-types A and F, respectively.

3.2 Leave-one-out

The leave-one-out cross-validation technique has been applied to the codification variants for Biodiversity and HIV datasets (see Supplementary material for details about Kohonen and GCS models, quality error measures and SOM training parameters used). Since the Biodiversity dataset presents a high volume of gaps in its 32 sequences (22% per sequence, on average), it becomes especially appropriate to study the gap-coding factor.

Within Biodiversity dataset, for each of the six coding variants, 32 Kohonen networks with a 6×6 size in the output layer, and 32 GCS networks with three clusters of neurons in the output layer have been trained (see Supplementary material for details about leave-one-out results; Supplementary Fig S5; Supplementary Tables S5 and S6).

The weight associated to gaps is the only difference between *4DNear* and *4DFar* coding variants (Supplementary Table S1). Cross validation performed on the Biodiversity dataset with Kohonen and GCS networks showed that the weight gain of the gap incorporated in the *4DFar* variant does not improve error outcomes in comparison with *4DNear* variants. Since this result was obtained with a dataset showing a high percentage (22%) of gaps in the aligned sequences, the leave-one-out technique has been applied to the HIV dataset (with 0.4% gaps) using only the *3D* and *4DNear* codifications. In this case, the focus has been put on the comparison between the regular vs. irregular tetrahedron.

For each of the four coding variants used with the HIV dataset, 44 Kohonen networks with a 9×9 size in the output layer, and 44 GCS networks with 6 clusters of neurons in the output layer have been trained (see Supplementary material for details about leave-one-out results, as well as Supplementary Fig S6; Supplementary Tables S7 and S8). Due to the small number of sequences belonging to the HIV-1 groups O and N within the dataset, the optimal number of clusters obtained for GCS networks is smaller than the number of groups plus sub-types in the set (see Supplementary Table S3).

The following common results within both datasets must be highlighted: (i) similar quality error measurements have been obtained using both types of networks, Kohonen and GCS, for all the coding variants; (ii) the comparison of variants based on each of the tetrahedron symmetries shows that both quality error measurements yield better results for *4D* variants with respect to their corresponding *3D* ones, despite the low percentage of gaps in HIV dataset; (iii) the *3DIrreg* variant improves Hamming error measures compared with the *3DReg*.

The analysis of the classification accuracy within Biodiversity dataset showed that the sequence not used in the training of the network has always been correctly classified according to its phylogenetic domain (see Supplementary Tables S5 and S6). Furthermore, with HIV dataset the percentage of sequences correctly classified within their phylogenetic group (M, N and O) is 100% in all cases except for *4DRegNear* variant using GCS networks (97.73%) (see Supplementary Tables S7–S9). In both datasets, the mapping of most of the sequences not used in training is highly consistent, and such a consistency could be further improved by including a higher number of sequences. In most cases, cross validation with GCS networks has thrown classification success rates slightly lower than Kohonen networks (see Supplementary Tables S5–S8). This fact may be due to the compact clustering performed by this type of network that generally stores inherent knowledge of the training sequences in a smaller volume of neurons than Kohonen network.

The evaluation of the codification factor affecting the nucleotide substitution mutations showed that, on the one hand, the comparison of *4DRegNear* and *4DIrrNear* variants in both datasets rendered similar results with respect to Hamming distances, while *4DRegNear* improved numeric error measures. Conversely, by comparing *3D* variants, the codification based on an irregular tetrahedron (*3DIrreg*) produced better Hamming distances (2% and 0.1–2% improvements in Biodiversity and HIV datasets, respectively) but slightly higher values in numeric error measures. The *4D* variant that rendered the best results (*4DRegNear*) compared with the best *3D* variant (*3DIrreg*) improved in 2–4% Hamming distance measures and also showed better mean numeric error results. However, it should not be neglected that any *4D* codification generates longer vectors ($4 \times L$) than those obtained with *3D* encoding variants ($3 \times L$), which subsequently affects the processing of such vectors both in terms of memory space and computational time. The SOM training time is directly correlated to the length and the number of training vectors. If the training dataset has many sequences with many nucleotides, it may be impossible to hold them in memory, and they should be read from disc at each learning iteration. Considering that reading a sequence from disc is several orders of magnitude slower than getting from memory (about 200 000 times higher), any improvement in the training time is desirable. For the same number of training iterations and neurons in the network, the training time relationship between *3D* versus *4D* codification variants is 3/4. If either vector size or computation speed becomes critical factor (e.g. when much longer sequences or a higher number of them are to be classified), *3DIrreg* would be the best choice among the six variants proposed for encoding DNA sequences.

3.3 Distribution of sequences by means of SOM

After applying the leave-one-out methodology, several SOMs have been trained using the complete set of sequences of Biodiversity and HIV. In Kohonen networks, the higher the learning rate (α_1) the higher the final dispersion of sequences is, although this decreases the topology preservation of the network [measured using the Kaski-Lagus function (Kaski and Lagus, 1996)]. Values of learning rate in the range [0.1, 0.2] get a right balance between both parameters. In this section, Kohonen networks have been trained with the

extreme values of this range to analyse the influence of the learning rate in the level of taxonomic resolution and topology preservation of the networks. For each of the trained networks, the U-matrix is calculated (see SOM visualization on Supplementary material), and all the training sequences are mapped and labeled over the neuron that best matches with the pattern in Euclidean distance (the so-called 'best matching unit', bmu).

Four Kohonen SOMs with a 6×6 size and two GCS networks with three clusters of neurons in the output layer have been trained using the 32 SSU rRNA sequences of the Biodiversity dataset codified by *4DRegNear* and *3DIrreg* variants. The labeled U-matrix obtained for the six networks is shown in Figure 1. Outlined circles represent the neurons, gray-scale tone inside the circle indicates the mean Euclidean distance between the prototype vector of the neuron and its immediate neighbors, and the gray tone of the circles without outline placed between two neighboring neurons identifies the Euclidean distance between both prototype vectors (where black means little distance, and white large distance). For the sake of clarity, we arranged the labels of the sequences around the map and joint each group with its bmu by a line. The six U-matrices depicted in Figure 1 show the correct mapping of the 32 sequences of Biodiversity dataset within the phylogenetic domain to which they belong (see Supplementary Fig S3). The four Kohonen networks trained with the two learning rate values clearly display the higher sequence dispersion effect when $\alpha_1 = 0.2$ is used (Fig. 1A versus C and B versus D), while the topographic error measured with Kaski–Lagus function grows from 13.71 to 21.61 in the *4DRegNear* variant and from 19.24 to 23.36 within *3DIrreg*. Comparing the distribution of sequences obtained by the two Kohonen SOM trained using the learning rate $\alpha_1 = 0.1$, it is noteworthy that the one trained with the *3DIrreg* variant (Fig. 1B) maps the sequences in a more compact way than that trained with the *4DRegNear* (Fig. 1A), i.e. the number of neurons that map the 32 sequences is smaller. This two Kohonen networks properly discriminated the phyla *Crenarchaeota* and *Euryarchaeota* within Archaea domain, and the one trained using the variant *4DRegNear* (Fig. 1A) completely separated the classes within them (see Supplementary Table S2). However, despite the wider distribution of sequences in the network trained with *4DRegNear* variant (Fig. 1A), none of the two Kohonen networks allowed the complete discrimination of either the phyla *Proteobacteria* and *Cyanobacteria* within Bacteria or the kingdoms within Eukarya. However, the two Kohonen networks trained with the learning rate $\alpha_1 = 0.2$ showed a higher discrimination power (Fig. 1C and D), getting class-resolution within Archaea and Bacteria domains, and phylum-resolution within Eukarya. Regarding the GCS networks, both the one trained with the *4DRegNear* variant (Fig. 1E) and that trained with the *3DIrreg* (Fig. 1F) produced three clusters of neurons, unequivocally assigning the sequences to each of the three phylogenetic domains present in the dataset. In this case, the phyla *Crenarchaeota* and *Euryarchaeota* within Archaea domain are clearly distinguished, and the class *Halobacteria* was correctly classified within *Euryarchaeota*. However, both networks mapped classes *Methanobacteria* and *Thermoplasmata* on the same neuron (see Fig. 1E and F; Supplementary Table S2). In addition, the two GCS networks clearly discriminated the available phyla and classes within Bacteria. With regard to Eukarya domain, the network trained with *3DIrreg* variant grouped together all the kingdoms, though the one trained with *4DRegNear* variant was also able to distinguish the sequence representative of the kingdom *Metazoa* and that of the *Alveolata* (see Fig. 1E and F; Supplementary Table S2).

In turn, using the 44 sequences of the HIV dataset codified by *4DRegNear* and *3DIrreg* variants, four Kohonen SOMs with a 9×9 size and two GCS networks with six clusters of neurons in the output layer have been trained. Figure 2 shows the labeled U-Matrix obtained for the six networks. Once again, the two Kohonen networks trained with a higher learning rate ($\alpha_1 = 0.2$) achieved greater dispersion of sequences although, again, the topographic error measured with Kaski–Lagus function grows from 5.80 to 7.62 in the *4DRegNear* variant and from 7.61 to 10.28 within *3DIrreg*. In networks trained using *4DRegNear* variant, the sequences have been correctly classified according to their group, sub-type and sub-subtype (see Supplementary Fig S4, and Fig. 2A, C and E). Furthermore, those trained with *3DIrreg* variant discriminated group and subtype, but only the Kohonen network trained with $\alpha_1 = 0.2$ distinguishes the sub-subtypes A1 and A2 (see Fig. 2B, D and F). In the four Kohonen networks, sequences belonging to the group O were clearly distinguished in a corner of the map (Fig. 2A–D), forming the cluster of neurons that showed a greater distance to the rest of clusters. Furthermore, within group M, the sub-types B and D were identified by neurons placed very close to each other in the map, and this was also the case for sub-types F and K. This result fully agrees with the phylogenetic relationship among such HIV-1 sub-types (see Supplementary Fig S4). In the two GCS networks, a cluster of neurons identifies the sequences belonging to the group O, while other cluster groups sub-types B and D (Fig. 2E and F). However, only the network trained with the *3DIrreg* variant grouped F and K sub-types in the same cluster, thus showing the best discrimination power with this dataset.

3.4 Comparison with binary codification method

The codification method proposed in this work transforms aligned symbolic DNA sequences into numeric data and vice versa, and it could be considered as one the fixed mapping methods (Kwan and Arniker, 2009). Some of these methods introduce mathematical properties in the resulting numeric vectors, which do not correspond to any feature of the DNA sequence. In turn, the variants proposed in this work use different Euclidean distances to weight DNA mutations due to nucleotide substitutions and 'indels'. Another drawback of the DNA encoding methods is the impossibility to deal with symbols corresponding to ambiguous nucleotides and gaps (Kwan and Arniker, 2009). The 'binary' DNA codification is an alternative bijective fixed mapping method that avoids these restrictions (Andrade *et al.*, 1997; Nantasenamat *et al.*, 2009). In 'binary' codification the four nucleotides are represented using a probabilistic encoding (A{1,0,0,0}, G{0,1,0,0}, C{0,0,1,0}, T{0,0,0,1}), with four numerical values for each nucleotide. This method considers both the gaps resulting of the sequences alignment and the ambiguous nucleotides. To codify the gap, the four numerical components are set to zero ({0,0,0,0}), and probability values in the corresponding positions are used to codify ambiguous nucleotides, e.g. R is coded as {0.5, 0.5, 0, 0}, B as {0, 0.33, 0.33, 0.33} and N as {0.25, 0.25, 0.25, 0.25}.

To compare *4DRegNear* and *3DIrreg* variants with this 'binary' DNA codification, 50 Kohonen networks have been trained for each dataset and codification method, and three error measures have been calculated: the mean of the average quantization error, Qe, the mean of the Kaski–Lagus function (Kaski and Lagus, 1996), ε_{kl}, and the mean of the average Hamming distance between the training sequences and their bmu's previously transformed to DNA format, Hm% (see Supplementary Material for details about these mean error measures). The values of Qe and ε_{kl} evaluate how well the

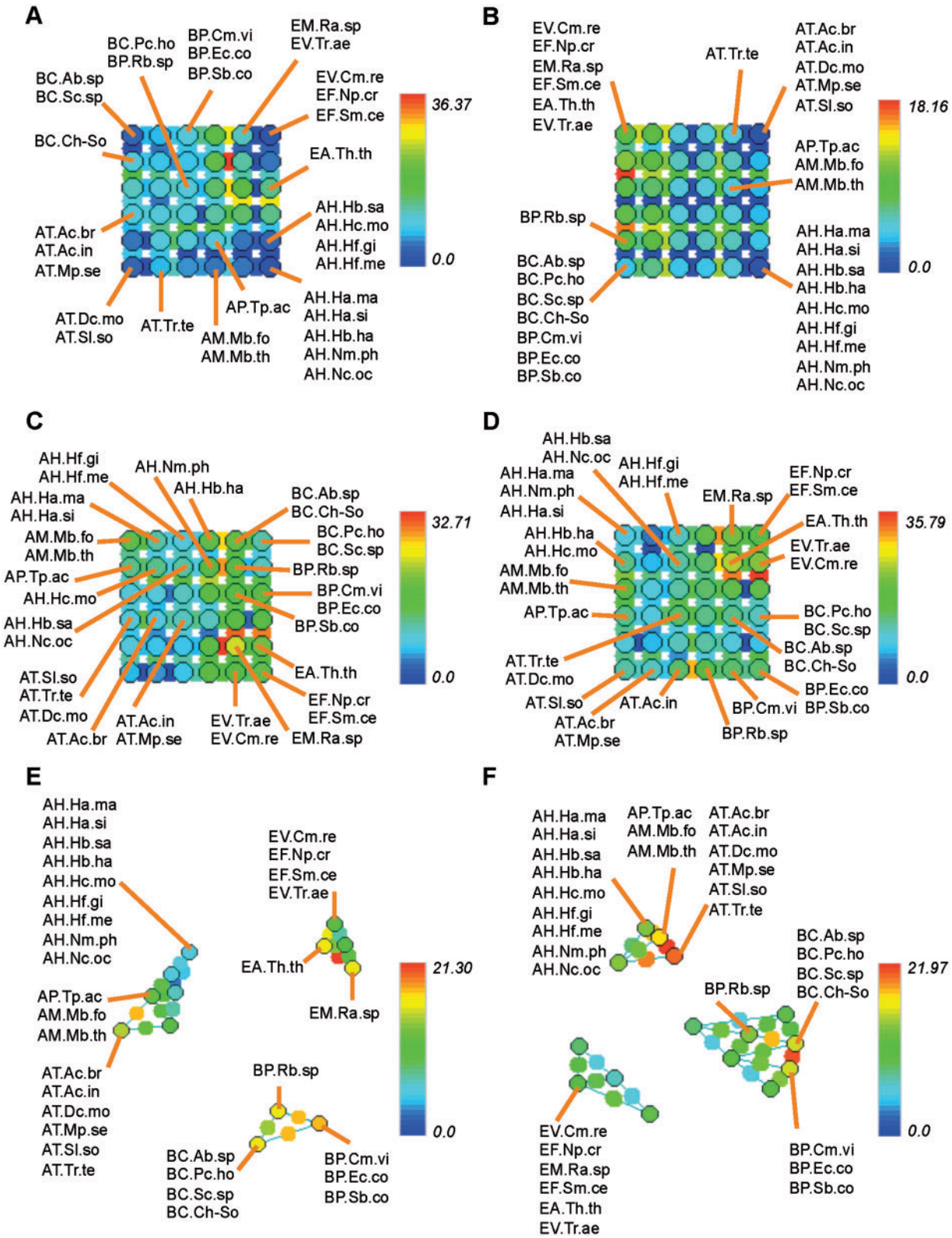

Fig. 1. Distribution of sequences performed by SOM for Biodiversity dataset. Labeled U-matrix. (**A**) 6×6 Kohonen SOM trained with *4DRegNear* variant and $\alpha_1 = 0.1$ learning rate. (**B**) 6×6 Kohonen SOM trained with *3DIrreg* variant and $\alpha_1 = 0.1$ learning rate. (**C**) 6×6 Kohonen SOM trained with *4DRegNear* variant and $\alpha_1 = 0.2$ learning rate. (**D**) 6×6 Kohonen SOM trained with *3DIrreg* variant and $\alpha_1 = 0.2$ learning rate. (**E**) Three clusters GCS network trained with *4DRegNear* variant. (**F**) Three clusters GCS network trained with *3DIrreg* variant

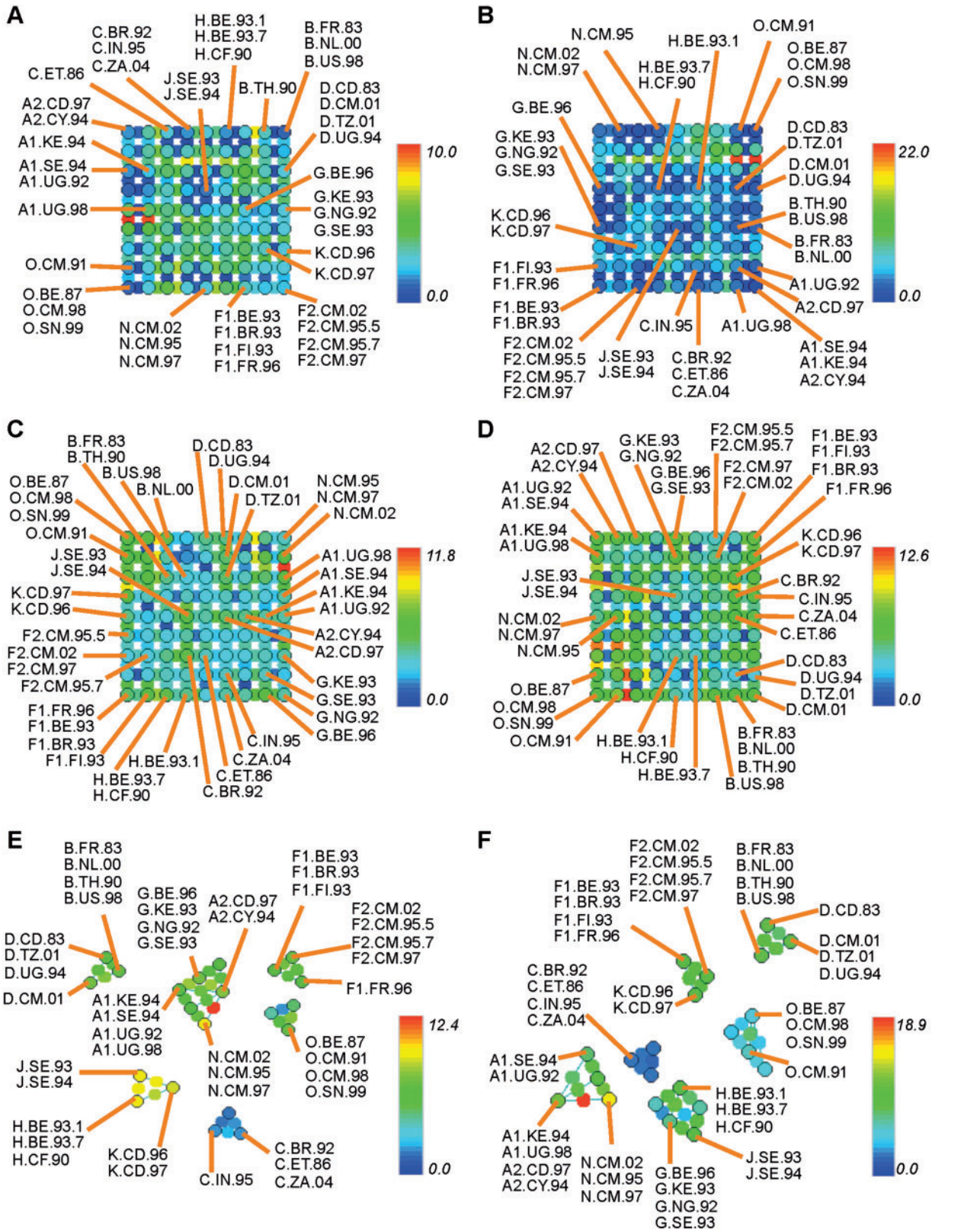

Fig. 2. Distribution of sequences performed by SOM for HIV dataset. Labeled U-matrix. (**A**) 9×9 Kohonen SOM trained with *4DRegNear* variant and $\alpha_1 = 0.1$ learning rate. (**B**) 9×9 Kohonen SOM trained with *3DIrreg* variant and $\alpha_1 = 0.1$ learning rate. (**C**) 9×9 Kohonen SOM trained with *4DRegNear* variant and $\alpha_1 = 0.2$ learning rate. (**D**) 9×9 Kohonen SOM trained with *3DIrreg* variant and $\alpha_1 = 0.2$ learning rate. (**E**) Six clusters GCS network trained with *4DRegNear* variant. (**F**) Six clusters GCS network trained with *3DIrreg* variant

SOM network is adapted to the numeric input data (known as topographic error measures), and Hm% indicates the accuracy of the SOM map in representing the input space expressed on DNA sequence format. The size of the trained networks was the same than that used in the previous sections: 6×6 for Biodiversity dataset, and 9×9 for HIV dataset (with $\alpha_1 = 0.2$, to achieve a greater dispersion of the sequences).

The proposed *4DRegNear* codification variant rendered the best results in topographic error measures (Qe and ε_{kl}) and average Hamming distance (Hm%) for both datasets (Supplementary Fig S7). Conversely, comparing 'binary' codification with the proposed *3DIrreg* variant the former has produced better results in average Hamming distance using both datasets, although with the HIV *3DIrreg* variant got slightly better topographic error measures (Supplementary Fig S7B).

Among the group of 50 Kohonen SOMs trained with the Biodiversity dataset and 'binary' codification, one showing error values similar to the mean values Qe, ε_{kl} and Hm% has been selected and the labeled U-matrix has been obtained (Fig. 3A). This SOM gets class-resolution within Archaea and Bacteria domains, but it does not discriminate kingdoms within Eucarya (see Supplementary Fig S3). In this sense, the two Kohonen SOMs trained with *4DRegNear* and *3DIrreg* variants (and $\alpha_1 = 0.2$) showed a higher discrimination power (Fig. 1C and D).

Figure 3B shows the labeled U-matrix associated with a Kohonen SOMs trained with HIV dataset using 'binary' codification (one with error values similar to the mean values Qe, ε_{kl} and Hm%). In this case, the sequences have been correctly classified according to their group, sub-type, and sub-subtype (see Supplementary Fig S4), the same taxonomic resolution as the two Kohonen SOMs trained with *4DRegNear* and *3DIrreg* variants and $\alpha_1 = 0.2$ (Fig. 2C and D).

For DNA sequences of L nucleotides long, the 'binary' method and the *4DRegNear* variant produce numeric vectors of the same length ($4 \times L$). In this sense, both codifications provide the same SOM training conditions, but the proposed *4DRegNear* variant improves the adaptation measures of the SOM to the input space with respect to the 'binary' codification. Furthermore, the proposed *3DIrreg* variant produces shorter numeric vectors ($3 \times L$). Although the adaptation measures obtained with this variant are slightly worse that those obtained with 'binary' codification, *3DIrreg* variant improves the memory requirements and the SOM training time in a ratio of 3/4. Therefore, both *4DRegNear* and *3DIrreg* codification variants show clear, though distinct, advantages over the traditionally used 'binary' method.

4 Conclusions

In this work, a new bijective technique for coding DNA and RNA sequences has been developed and tested. The proposed method, based on the assignment of the nucleotides to the coordinates of the vertices of either a 3D or a 4D tetrahedron, facilitates the codification of ambiguous nucleotides just using geometrical distances between two, three or four nucleotides. The transformation of sequences to numeric vectors has proven to be useful for neural networks training, as it is the case of Kohonen's SOM and GCS. The bijective property of the codification method allows coding the sequences to numeric coordinates and, vice versa, transforming the prototype vectors to their nucleotide sequence format. The possibility of working in the space of either sequences or numeric vectors is one of the advantages of the codification method proposed here. Then, the error values can be measured either as vectorial (Euclidean) or Hamming distances. Error measure by means of Hamming distance, as used in Section 3, has special relevance when the numeric error measures are not particularly significant.

The SOM neural network model provides a set of relevant advantages for processing nucleotide sequences, including: (i) it is an unsupervised method, thus knowing the class associated to the sequences is not required; (ii) SOM projects a high-dimensional input space over a low dimensional one, thus producing several 2D graphs used to analyse some features of the sequences (e.g. number of clusters, or similarity and distances between sequences); (iii) once the network has been trained, new sequences can be projected, and the relationship with those used in the training can be displayed. On the contrary, the construction of a phylogenetic tree requires the use of all the sequences with which a new sequence has to be compared.

Two SOM models, Kohonen's SOM and GCS, and two distinct sets of sequences, Biodiversity and HIV, have been used for testing the codification variants. The results presented here show that the two SOM methods distribute the analysed sequences in agreement with their taxonomic diversity and phylogenetic relationships. With Biodiversity dataset, GCS networks were able to discriminate up to phyla level within Archaea domain, and they got classes resolution

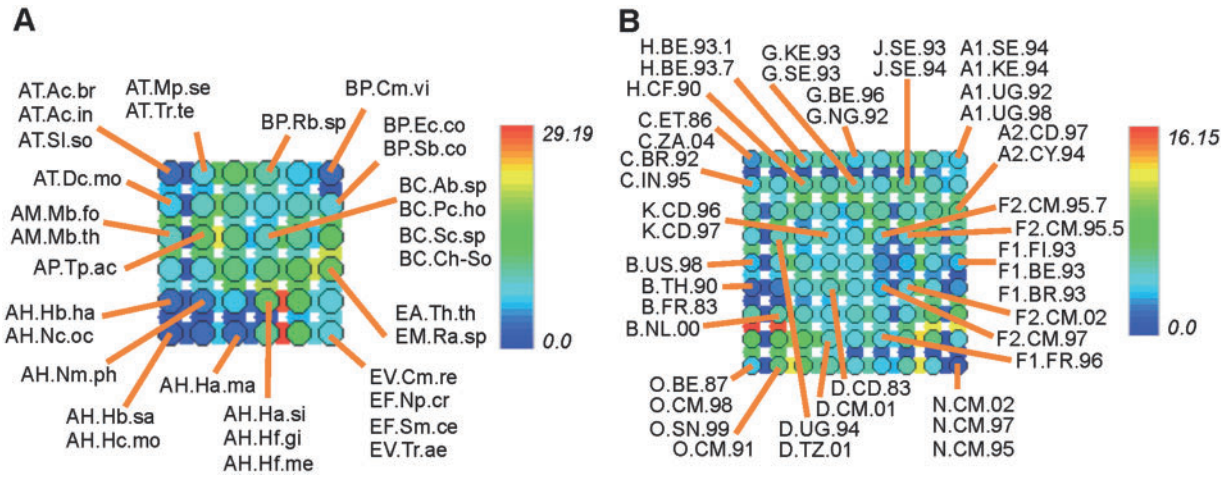

Fig. 3. Distribution of sequences performed by SOM with binary codification. (**A**) 6×6 Kohonen SOM trained with Biodiversity dataset. (**B**) 9×9 Kohonen SOM trained with HIV dataset

in Bacteria. Conversely, Kohonen SOMs trained with a learning rate $\alpha_1 = 0.2$ discriminated up to class level within Archaea and Bacteria, and they reached the phyla level for Eukarya. With HIV dataset, both Kohonen and GCS models achieved group and sub-type resolution, and sub-subtype information was retrieved. Generally, in GCS model the knowledge is stored in a compact way compared with Kohonen SOMs, because they typically use fewer neurons. That is why the projection of the sequences on the GCS maps usually shows lower dispersion that that of the Kohonen networks. In addition, GCS networks can automatically cluster the training sequences, thus producing a cluster of neurons by type if enough training sequences are available.

Six variants of the method for coding two different sets of nucleotide sequences have been compared. Leave-one-out experiments with both datasets have shown that *4DRegNear* variant provides the best results in error measurements, independently of the percentage of alignment-derived gaps present in each dataset. Regarding the error measurements of the two *3D* variants, *3DIrreg* codification has produced better results than *3DReg* variant, both for Biodiversity and HIV datasets.

The *4DRegNear* and *3DIrreg* variants, based on Euclidean space, as SOM processing, have been compared with the bijective 'binary' codification method, based on probabilistic values. The results have shown that *4DRegNear* improves the SOM adaptation measures, while *3DIrreg* provides a faster SOM training than the 'binary' method. In addition, the use of *4DRegNear* and *3DIrreg* variants showed better or equal taxonomic resolutions than the 'binary' method. Therefore, our results demonstrate that the most important factor affecting the accuracy of sequence classification is the assignment of an extra weight to the presence of alignment-derived gaps. In turn, if gaps either do not exist or are not considered, the classification benefits from weighting transversion:transition ratio as 2:1.

Based on the successful results of this study, future work will include large-scale application of the SOM methods using the *4DRegNear* and *3DIrreg* codification variants to process and topologically classify large sets of nucleic acid sequences.

Funding

This work is supported by Spanish Ministry of Economy and Competitiveness [BFU2012-39816-C02-02 to F.M. and BIO2013-47228-R to C.B.], Consolider/Ingenio2010 (CSD2007-0002 MICINN to F.M.), Spanish Ministry of Science and Innovation [TIN2011-28627-C04-02 (ANYSELF) to J.J.M], CEI-BioTIC Granada [CEI2013-P-14 to J.J.M], GENIL PYR2014-17 to A.M.], Spanish CSIC [200920I040 to C.B.] and Spanish MICINN [BIO2010-20696 to C.B.]

Conflict of interest: none declared.

References

Afreixo,A. *et al*. (2009) Genome analysis with inter-nucleotide distances. *Bioinformatics*, **25**, 3064–3070.

Almeida,J.S. and Vinga,S. (2009) Biological sequences as pictures—a genetic two dimensional solution for iterated maps. *BMC Bioinformatics*, **10**:100.

Andrade,M. *et al*. (1997) Classification of protein families and detection of the determinant residues with an improved self-organizing map. *Biol. Cybern.*, **76**, 441–450.

Astel,A. *et al*. (2007) Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets. *Water Res.*, **41**, 4566–4578.

Boyle,A.P. *et al*. (2014) Comparative analysis of regulatory information and circuits across distant species. *Nature*, **512**, 453–456.

Briones,C. *et al*. (2005) Reconstructing evolutionary relationships from functional data: a consistent classification of organisms based on translation inhibition response. *Mol. Phylogenet. Evol.*, **34**, 371–381.

Chavez-Alvarez,R. *et al*. (2014) Discovery of possible gene relationships through the application of self-organizing maps to DNA microarray databases. *PLoS One*, **9**, e93233.

Delgado,S. *et al*. (2011) A combined measure for quantifying and qualifying the topology preservation of growing self-organizing maps. *Neurocomputing*, **74**, 2624–2632.

Domingo,E. *et al*. (2012) Viral quasispecies evolution. *Microbiol. Mol. Biol. Rev.*, **76**, 159–216.

Dozono,H. (2014) Visualization and classification of DNA sequences using pareto learning self organizing maps based on frequency and correlation coefficient. *Adv. Intell. Syst. Comput.*, **295**, 89–98.

Fankhauser,N. and Mäser,P. (2005) Identification of GPI anchor attachment signals by Kohonen self-organizing map. *Bioinformatics*, **21**, 1846–1852.

Fritzke,B. (1994) Growing cell structures—a self-organizing network for unsupervised and supervised learning, *Neural Netw.*, **7**, 1441–1460.

Jiang,X. *et al*. (2004) Median strings: a review. Data Mining in time series databases. *World Sci.*, **57**, 173–192.

Kaski,S. and Lagus,K. (1996) Comparing self-organizing maps. *Intl. Conf. Artif. Neural Netw. (ICANN)*, Springer, Berlin. 809–814.

Kwan,H.K. and Arniker,S.B. (2009) Numerical representation of DNA sequences. In: Proceeding of IEEE International conference of Electro/Information Technology, Windsor, Canada, 307–310.

Kimura,M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, **16**: 111–120.

Kohonen,T. (2001) *Self-Organizing Maps*. 3th edn. Springer, Berlin Heidelberg.

Kohonen,T. and Somervuo,P. (2002) How to make large self-organizing maps for nonvectorial data. *Neural Netw.*, **15**, 945–952.

Lo,N-W. *et al*. (2007) Global visualization and comparison of DNA sequences by use of three-dimensional trajectories. *J. InforSci. Eng.*, **23**, 1723–1736.

Murtagh,F. and Hernandez-Pajares,M. (1995) The Kohonen self-organizing map method: an assessment. *J. Classific.*, **12**, 165–190.

Nantasenamat,C. *et al*. (2009) A practical overview of quantitative structure-activity relationship. *EXCLI J.*, **8**, 74–88.

Pearson,W. and Lipman,D. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.

Sapp,J. (2009). *The New Foundations of Evolution: On the Tree of Life*. Oxford University Press, Inc., New York, p. 425.

Solnon,C. and Jolion,J.M. (2007) Generalized vs set median strings for histograms-based distances: algorithms and classification results in the image domain. *LNCS*, **4538**, 404–414.

Vesanto,J. and Alhoniemi,E. (2000) Clustering of the self-organizing map. *IEEE Trans. Neural Netw.*, **11**, 586–600.

Woese,C.R. and Fox,G.E. (1977) Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proc. Natl. Acad. Sci. USA*, **74**, 5088–5090.

Xu,R. and Wunsch,D. (2005) Survey of clustering algorithms. *IEEE Trans. Neural Netw.,* **16**, 645, 678.