On-line relational SOM for dissimilarity data

Madalina Olteanu, Nathalie Villa-Vialaneix, and Marie Cottrell

SAMM-Université Paris 1 Panthéon Sorbonne
90, rue de Tolbiac, 75013 Paris - France
{madalina.olteanu,nathalie.villa,marie.cottrell}@univ-paris1.fr
http://samm.univ-paris1.fr

Abstract. In some applications and in order to address real world situations better, data may be more complex than simple vectors. In some examples, they can be known through their pairwise dissimilarities only. Several variants of the Self Organizing Map algorithm were introduced to generalize the original algorithm to this framework. Whereas median SOM is based on a rough representation of the prototypes, relational SOM allows representing these prototypes by a virtual combination of all elements in the data set. However, this latter approach suffers from two main drawbacks. First, its complexity can be large. Second, only a batch version of this algorithm has been studied so far and it often provides results having a bad topographic organization. In this article, an on-line version of relational SOM is described and justified. The algorithm is tested on several datasets, including categorical data and graphs, and compared with the batch version and with other SOM algorithms for non vector data.

1 Introduction

In many real-world applications, data cannot be described by a fixed set of numerical attributes. This is the case, for instance, when data are described by categorical variables or by relations between objects (i.e., persons involved in a social network). A common solution to address this kind of issue is to use a measure of resemblance (i.e., a similarity or a dissimilarity) that can handle categorical variables, graphs or focus on specific aspects of the data, designed by expertise knowledge. Many standard methods for data mining have been generalized to non vectorial data, recently including prototype-based clustering. The recent paper [6] provides an overview of several methods that have been proposed to tackle complex data with neural networks.

In particular, several extensions of the Self-Organizing Maps (SOM) algorithm have been proposed. One approach consists in extending SOM to categorical data by using a method similar to Multiple Correspondence Analysis, [5]. Another approach uses the median principle which consists in replacing the standard computation of the prototypes by an approximation in the original dataset. This principle was used to extend SOM to dissimilarity data in [15]. One of the main drawbacks of this approach is that forcing the prototypes to be chosen

among the dataset is very restrictive; in order to increase the flexibility of the representation, [3] propose to represent a class by several prototypes, all chosen among the original dataset. However this method increases the computational time and prototypes still stay restricted to the original dataset, hence reflecting possible sampling or sparsity issues.

An alternative to median-based algorithms relies on a method that is close to the classical algorithm used in the Euclidean case and is based on the idea that prototypes may be expressed as linear combinations of the original dataset. In the kernel SOM framework, this setting is made natural by the use of the kernel that maps the original data into a (large dimensional) Euclidean space (see [16, 1] for on-line versions and [2] for the batch version). Many kernels have been designed to handle complex data such as strings, nodes in a graphs or graphs themselves [10].

More generally, when the data are already described by a dissimilarity that is not associated to a kernel, [12, 18, 11] use a similar idea. They introduce an implicit "convex combination" of the original data to extend the classical batch versions of SOM to dissimilarity data. This approach is known under the name "relational SOM". The purpose of the present paper is to show that the same idea can be used to define on-line relational SOM. Such an approach reduces the computational cost of the algorithm and leads to a better organization of the map. In the remaining of this article, Section 2 describes the methodology and Section 3 illustrates its use on simulated and real-world data.

2 Methodology

In the following, let us suppose that n input data, $x_1, \ldots, x_n$, from an arbitrary input space $\mathcal{G}$ are given. These data are described by a dissimilarity matrix $\mathbf{D} = (\delta_{ij})_{i,j=1,\ldots,n}$ such that D is non negative ($\delta_{ij} \geq 0$), symmetric ($\delta_{ij} = \delta_{ji}$) and null on the diagonal ($\delta_{ii} = 0$). The purpose of the algorithm is to map these data into a low dimensional grid composed of U units which are linked together by a neighborhood relationship $K(u, u')$. A prototype p_u is associated with each unit $u \in \{1, \ldots, U\}$ in the grid. The U prototypes $(p_1, p_2, \ldots, p_U)$ are initialized either randomly among the input data or as random convex combinations of the input data.

In the Euclidean framework, where the input space is equipped with a distance, the matrix D is the distance matrix with entries $\delta_{ij} = \|x_i - x_j\|^2$. In this case, the on-line SOM algorithm iterates

- an *assignment step*: a randomly chosen input x_i is assigned to the closest prototype denoted by $p_{f(x_i)}$ according to shortest distance rule

$$f(x_i) = \arg \min_{u=1,\ldots,U} \|x_i - p_u\|,$$

- a *representation step*: all prototypes are updated

$$p_u^{\text{new}} = p_u^{\text{old}} + \alpha K(f(x_i), u)\,(x_i - p_u)\,,$$

where α is the training parameter.

In the more general framework, where the data are known through pairwise distances only, the assignment step cannot be carried out straightforwardly since the distances between the input data and the prototypes may not be directly computable. The solution introduced in [18] consists in supposing that prototypes are convex combinations of the original data, $p_u = \sum_i \beta_{ui} x_i$ with $\beta_{ui} > 0$ and $\sum_i \beta_{ui} = 1$. If β_u denotes the vector $(\beta_{u1}, \beta_{u2}, \ldots, \beta_{un})$, the distances in the assignment step can be written in terms of D and β_u only:

$$\|x_i - p_u\|^2 = (D\beta_u)_i - \frac{1}{2}\beta_u^T D \beta_u.$$

According to [18], the equation above still holds if the matrix D is no longer a distance matrix, but a general dissimilarity matrix, as long as it is symmetric and null on the diagonal. A generalization of the batch SOM algorithm, called batch relational SOM, which holds for dissimilarity matrices is introduced in [18].

The representation step may also be carried out in this general framework as long as the prototypes are supposed to be convex combinations of the input data. Hence, using the same ideas as [18], we introduce the on-line relational SOM, which generalizes the on-line SOM to dissimilarity data. The proposed algorithm is the following:

Algorithm 1 On-line relational SOM

1: For all $u = 1, \ldots, U$ and $i = 1, \ldots, n$, initialize β_{ui}^0 randomly in $\mathbb{R}$, such that $\beta_{ui}^0 \geq 0$ and $\sum_i^n \beta_{ui}^0 = 1$.
2: **for** t=1,...,T **do**
3: Randomly chose an input x_i
4: *Assignment* : find the unit of the closest prototype

$$f^t(x_i) \leftarrow \arg \min_{u=1,\ldots,U} \left(\beta_u^{t-1}\mathbf{D}\right)_i - \frac{1}{2}\beta_u^{t-1}\mathbf{D}(\beta_u^{t-1})^T$$

5: *Update of the prototypes*: $\forall\, u = 1, \ldots, U$,

$$\beta_u^t \leftarrow \beta_u^{t-1} + \alpha^t K^t(f^t(x_i), u)\left(\mathbf{1}_i - \beta_u^{t-1}\right)$$

where $\mathbf{1}_i$ is a vector with a single non null coefficient at the ith position, equal to one.
6: **end for**

In the applications of Section 3, the parameters of the algorithm are chosen according to [4]: the neighborhood K^t decreases in a piecewise linear way, starting from a neighborhood which corresponds to the whole grid up to a neighborhood restricted to the neuron itself; α^t vanishes at the rate of $1/t$. Let us remark that if the dissimilarity matrix is a Euclidean distance matrix, relational on-line

SOM is equivalent to the classical on-line SOM algorithm, as long as the n input data contain a basis of the input space $\mathcal{G}$.

As explained in [8], although batch SOM possesses the nice properties of being deterministic and of usually converging in a few iterations, it has several drawbacks such as bad organization, bad visualization, unbalanced classes and strong dependence on the initialization. Moreover, the computational complexity of the online algorithm may be significantly reduced with respect to the batch algorithm. For one iteration, the complexity of the batch algorithm is $\mathcal{O}(Un^3 + Un^2)$, while for the online algorithm it is $\mathcal{O}(Un^2+Un)$. However, since the online algorithm has to scan all input data, the number of iterations is significantly larger than in the batch case. To summarize, if T_1 is the number of iterations for batch relational SOM and T_2 is the number of iterations for online relational SOM, the ratio between the two computation times will be $T_1 n/T_2$.

For illustration, let us consider 500 points sampled randomly from the uniform distribution in $[0, 1]^2$. The batch version of relational SOM and the on-line version of relational SOM were performed with identical 10x10 grid structures and identical initializations. Results are available in Figure 1. Batch relational SOM converged quickly, in 20 iterations (the grid organization is represented at iterations 0 (random initialization), 5, 9, 13, 17 and 20), but the map is not well organized. On-line relational SOM converged in less than 2500 iterations (the grid organization is represented at iterations 0 (initialization), 500, 1000, 1500, 2000 and 2500), but the map is now almost perfectly organized. This results was achieved in 40 minutes for the batch version and in 10 minutes for the on-line version on a netpc (with $2 \times$ 1GHz AMD processors and 4Go RAM).

3 Applications

This section presents several applications of the on-line relational SOM on various datasets. Section 3.1 deals with simulated data described by numerical variables, but organized on a non linear surface. Section 3.2 is an application on a real dataset where the individuals are described by categorical variables. Finally, Section 3.3 is an application to the clustering of nodes of a graph.

3.1 Swiss roll

Let us first use a toy example to illustrate the stochastic version of relational SOM. The simulated data is the popular Swiss roll, a two-dimensional manifold embedded in a three-dimensional space. This example has already been used for illustrating the performances of Isomap [20]. The data has the shape illustrated by Figure 2. 5 000 points were simulated. However, since all methods presented here work with matrices of pairwise distances, the computation times would have been rather heavy for 5 000 points. Hence, we run the different algorithms on 1 000 points uniformly distributed on the manifold. First, the distance matrix was computed using the geodesic distance based on the K-rule with $K = 10$. Then, two types of algorithms were performed: multidimensional scaling and

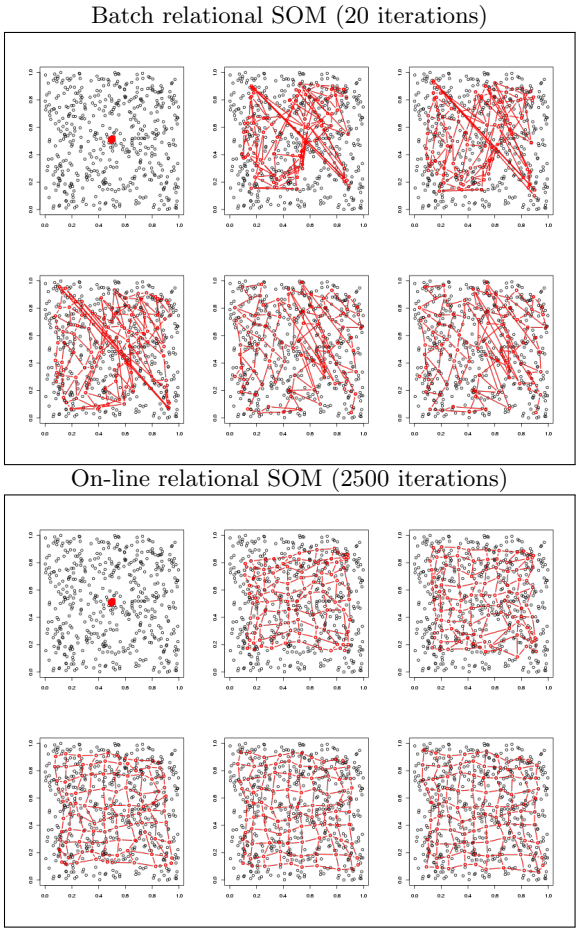

Fig. 1. Batch and on-line SOM organization for 500 samples from the uniform distribution in $[0, 1]^2$. The same initialization was used for both algorithms.

self-organizing maps. The results obtained with Isomap [20] are available in Figure 2. As expected, both methods succeed in unfolding the Swiss roll and the results are very similar. Next, batch median SOM and on-line relational SOM were applied to the dissimilarity matrix computed with the geodesic distance. As shown in Figure 3, the size of the map plays an important role in unfolding the data. For squared grids, the problem is not completely solved by either of the two algorithms. Nevertheless, on-line relational SOM manages to project the different scrolls of the roll into separate regions on the map. Moreover, some empty cells highlight the roll structure, which is not completely unfolded but rather projected without overlapping. Since squared grids appeared too heavily constrained, we also tested rectangular grids. The results are better for both

algorithms which both manage to unfold the data. However, the on-line version clearly outperforms the batch version.

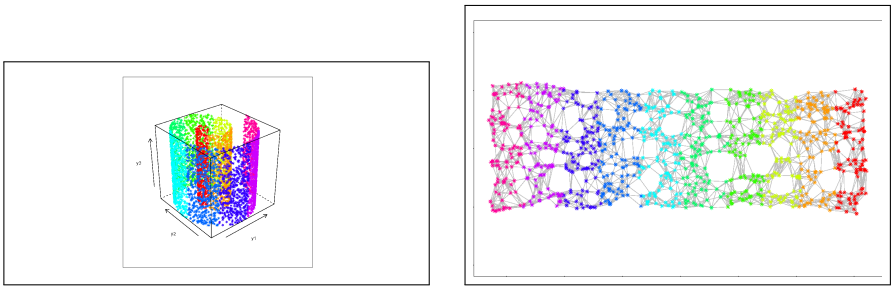

Fig. 2. Unfolding the Swiss roll using Isomap

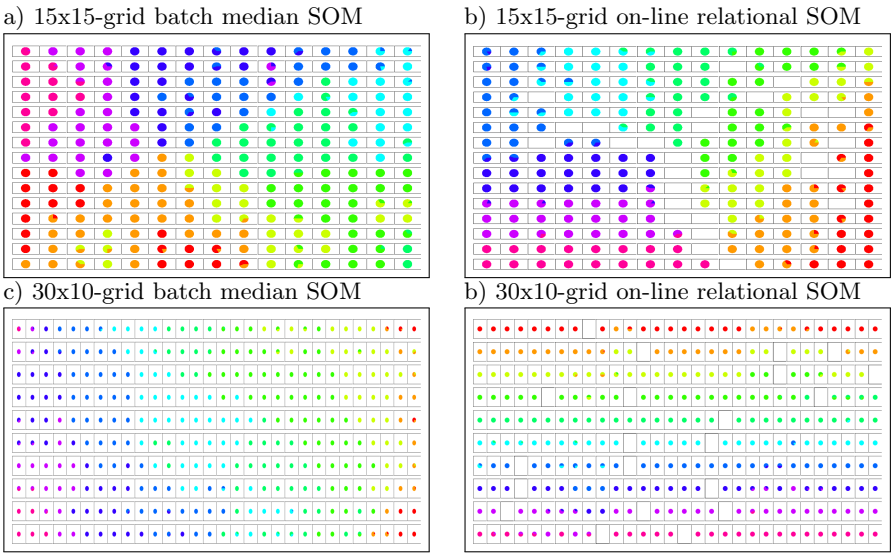

Fig. 3. Unfolding the Swiss roll using self-organizing maps

3.2 Amazonian butterflies

This data set contains 465 input data and was previously used by [13] to demonstrate the synergy between DNA barcoding and morphological-diversity studies. The notion of DNA barcoding comprises a wide family of molecular and bioinformatics methods aimed at identifying biological specimens and assigning them

to a species. According to the vast literature published during the past years on the topic, two separate tasks emerge for DNA barcoding: on the one hand, assign unknown samples to known species and, on the other hand, discover undescribed species, [7]. The second task is usually approached with the Neighbor Joining algorithm [19] which constructs a tree similar to a dendrogram. When the sample size is large, the trees become rapidly unreadable. Moreover, they are quite sensitive to the order in which the input data are presented. Let us also mention that unsupervised learning and visualization methods are used to a very limited extent by the DNA barcoding community, although the information they bring may be quite useful. The use of self-organizing maps may be quite helpful in visualizing the data and bringing out clusters or groups of clusters that may correspond to undescribed species.

DNA barcoding data are composed of sequences of nucleotides, i.e. sequences of "a", "c", "g", "t" letters in high dimension (hundreds or thousands of sites). Specific distances and dissimilarities such as the Kimura-2P ([14]) are usually computed. Hence, since the data is not Euclidean, dissimilarity-based methods appear to be more appropriate. Recently, batch median SOM was tested in [17] on several data sets, amongst which the Amazonian butterflies. Although median SOM provided encouraging results, two main drawbacks emerged. First, since the algorithm was run in batch, the organization of the map was generally poor and highly depending on the initialization. Second, since the algorithm calculates a prototype for each cluster among the dataset, it does not allow for empty clusters. Thus, the existence of species or groups of species was difficult to acknowledge. The use of on-line relational SOM overcomes these two issues. As shown in Figure 4, clusters are generally not mixing species, while the empty cells allow detecting the main groups of species. The only mixing class corresponds to a labeling error. Unsupervised clustering may thus be useful in addressing misidentification issues. In Figure 4b, distances with respect to the nearest neighbors were computed for each node. The distance between two nodes/cells is computed as the mean dissimilarity between the observations within each class. A polygon is drawn within each cell with vertices proportional to the distances to its neighbors. If two neighbor prototypes are very close, then the corresponding vertices are very close to the edges of the two cells. If the distance between neighbor prototypes is very large, then the corresponding vertices are far apart, close to the center of the cells.

3.3 Political books

This application uses a dataset modeled by a graph having 105 nodes. The nodes are books about US politics published around the time of the 2004 presidential election and sold by the on-line bookseller Amazon.com. Edges between two nodes represent frequent co-purchasing of the two books by the same buyers. The graph contains 441 edges and all nodes are labeled according to their political orientation (conservative, liberal or neutral). The graph has been extracted by Valdis Krebs and can be downloaded at `http://www-personal.umich.edu/~mejn/netdata/polbooks.zip`.

a) Species diversity (radius proportional to the size of the cluster)

b) Distances between prototypes

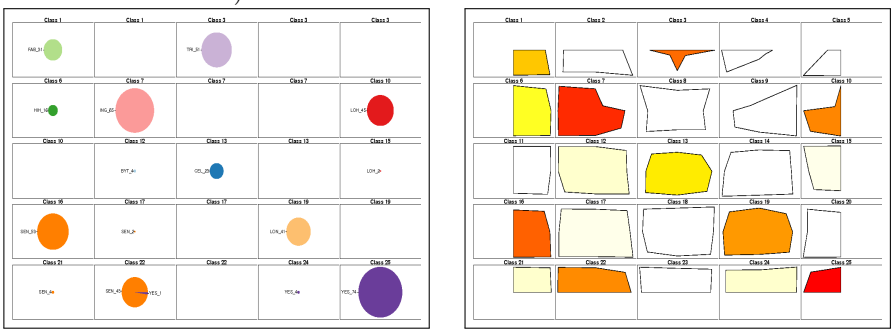

Fig. 4. On-line relational SOM for Amazonian butterflies

On-line relational SOM was used to cluster the nodes of the graph, according to the length of the shortest path between two nodes, which is a standard dissimilarity measure between nodes in a graph. Figures 5 and 6 (left) provide two representations of the "political books" network: the first one is the original graph displayed with a force directed placement algorithm, which is the one described in [9] and colored according to the clusters in which the nodes are classified. The second one is a simplified representation of the graph on the grid, where each node represents a cluster. The colors in the first figure and the density of edges in the second one shows that the clustering has a good organization on the grid, according to the graph structure: groups of nodes that are densely connected are classified in the same or in close clusters whereas groups of nodes that are not connected are classified apart.

Additionally, Figure 6 provides the distribution of the node labels inside each cluster for the obtained clustering (on the right hand part of the figure). Almost all clusters contain books having the same political orientation. Clusters that contain books with multiple political orientations are in the middle of the grid and include neutral books. Hence, this clustering can give a clue on a more subtle political orientation than the original labeling: for instance, liberal books from cluster 12 probably have a weaker commitment that those from clusters 1 or 2.

4 Conclusion

An on-line version of relational SOM is introduced in this paper. It combines the standard advantages of the stochastic version of the SOM (better organization and faster computation) with the relational SOM that is able to handle data described by a dissimilarity. The algorithm shows good performances in projecting data described either by numerical variables or by categorical variable, as well as in clustering the nodes of a graph.

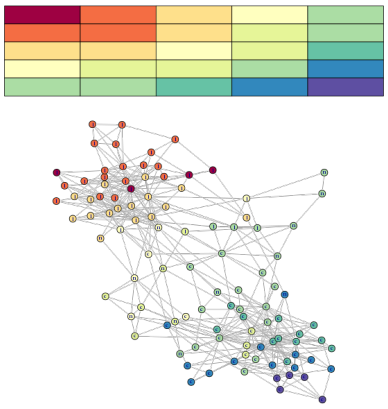

Fig. 5. "Political books" network displayed with a force directed placement algorithm. The nodes are labeled according to their political orientation and are colored according to a gradient that aims at emphasizing the distance between clusters on the grid, as represented at the top the figure.

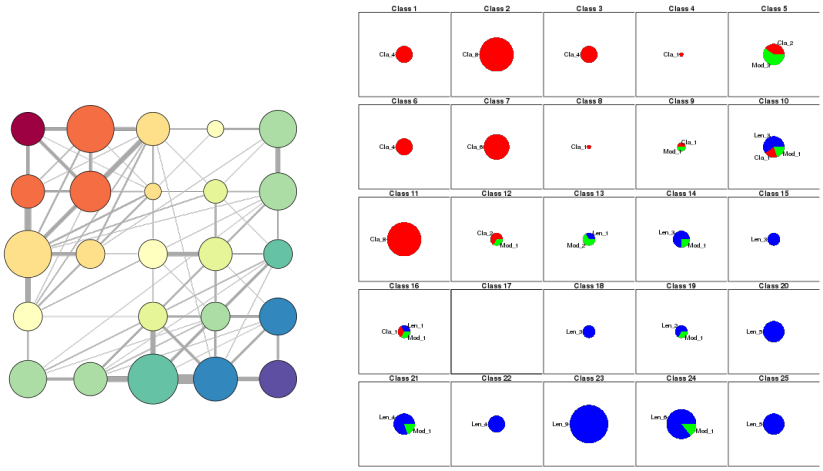

Fig. 6. Left: Simplified representation of the graph on the grid: each node represents a cluster whose area is proportional to the number of nodes included in it and the edges width represents the number of edges between the nodes of the corresponding cluster. Right: Distribution of the node labels for each neuron of the grid for the clustering obtained with the dissimilarity based on the length of the shortest paths. Red is for liberal books, blue for conservative books and green for neutral books.

References

1. Andras, P.: Kernel-Kohonen networks. International Journal of Neural Systems 12, 117–135 (2002)

2. Boulet, R., Jouve, B., Rossi, F., Villa, N.: Batch kernel SOM and related laplacian methods for social network analysis. Neurocomputing 71(7-9), 1257–1273 (2008)
3. Conan-Guez, B., Rossi, F., El Golli, A.: Fast algorithm and implementation of dissimilarity self-organizing maps. Neural Networks 19(6-7), 855–863 (2006)
4. Cottrell, M., Fort, J.C., Pagès, G.: Theoretical aspects of the SOM algorithm. Neurocomputing 21, 119–138 (1998)
5. Cottrell, M., Letrémy, P.: How to use the Kohonen algorithm to simultaneously analyse individuals in a survey. Neurocomputing 63, 193–207 (2005)
6. Cottrell, M., Olteanu, M., Rossi, F., Rynkiewicz, J., Villa-Vialaneix, N.: Neural networks for complex data. Künstliche Intelligenz 26(2), 1–8 (2012)
7. DeSalle, R., Egan, M., Siddal, M.: The unholy trinity: taxonomy, species delimitation and dna barcoding. Philosophical Transactions of the Royal Society B-Biological Sciences 360, 1905–1916 (2005)
8. Fort, J.C., Letremy, P., Cottrell, M.: Advantages and drawbacks of the batch kohonen algorithm. In: ESANN'02. pp. 223–230 (2002)
9. Fruchterman, T., Reingold, B.: Graph drawing by force-directed placement. Software-Practice and Experience 21, 1129–1164 (1991)
10. Gärtner, T.: Kernel for Structured Data. World Scientific (2008)
11. Hammer, B., Gisbrecht, A., Hasenfuss, A., Mokbel, B., Schleif, F., Zhu, X.: Topographic mapping of dissimilarity data. In: Proceedings of WSOM 2011. pp. 1–15 (2011)
12. Hammer, B., Hasenfuss, A., annd Strickert M. Rossi, F.: Topographic processing of relational data. In: Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07). Bielefeld, Germany (September 2007), to be published
13. Hebert, P.D.N., Penton, E.H., Burns, J.M., Janzen, D.H., Hallwachs, W.: Ten species in one: DNA barcoding reveals cryptic species in the neotropical skipper butterfly astraptes fulgerator. Genetic Analysis (2004)
14. Kimura, M.: A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. Journal of Molecular Evolution 16, 111–120 (1980)
15. Kohohen, T., Somervuo, P.: Self-Organizing maps of symbol strings. Neurocomputing 21, 19–30 (1998)
16. Mac Donald, D., Fyfe, C.: The kernel self organising map. In: Proceedings of 4th International Conference on knowledge-based intelligence engineering systems and applied technologies. pp. 317–320 (2000)
17. Olteanu, M., Nicolas, V., Schaeffer, B., Denys, C., Kennis, J., Colyn, M., Missoup, A.D., Laredo, C.: On the use of self-organizing maps for visualizing and studying barcode data. application to two data sets (2012), preprint submitted for publication
18. Rossi, F., Hasenfuss, A., Hammer, B.: Accelerating relational clustering algorithms with sparse prototype representation. In: 6th International Workshop on Self-Organizing Maps (WSOM). Neuroinformatics Group, Bielefeld University, Bielefeld, Germany (2007)
19. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4(4), 406–425 (1987), `http://mbe.oxfordjournals.org/content/4/4/406.abstract`
20. Tenenbaum, J.B., Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(5500), 2319–2323 (2000)