Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

Uncovering High-dimensional Structures of Projections from Dimensionality Reduction Methods

Michael C. Thrun, PhD[a,*], Alfred Ultsch, Prof. Dr. habil.[b]

[a] *Dept. of Hematology, Oncology and Immunology, Philipps-University of Marburg, Baldingerstraße, D-35043 Marburg*
[b] *Databionics Research Group, Philipps-University of Marburg, Hans-Meerwein-Straße 6, Marburg D-35032, Germany*

ABSTRACT

Projections are conventional methods of dimensionality reduction for information visualization used to transform high-dimensional data into low dimensional space. If the projection method restricts the output space to two dimensions, the result is a scatter plot. The goal of this scatter plot is to visualize the relative relationships between high-dimensional data points that build up distance and density-based structures. However, the Johnson–Lindenstrauss lemma states that the two-dimensional similarities in the scatter plot cannot coercively represent high-dimensional structures. Here, a simplified emergent self-organizing map uses the projected points of such a scatter plot in combination with the dataset in order to compute the generalized U-matrix. The generalized U-matrix defines the visualization of a topographic map depicting the misrepresentations of projected points with regards to a given dimensionality reduction method and the dataset.

- The topographic map provides accurate information about the high-dimensional distance and density based structures of high-dimensional data if an appropriate dimensionality reduction method is selected.
- The topographic map can uncover the absence of distance-based structures.
- The topographic map reveals the number of clusters in a dataset as the number of valleys.

ARTICLE INFO

Method name: Topographic Map Generation Using the Generalized Umatrix
Keywords: Dimensionality reduction, Projection methods, Data visualization, Unsupervised neural networks, Self-organizing maps
Article history: Received 22 January 2020; Accepted 4 October 2020; Available online 10 October 2020

DOI of original article: 10.1016/j.artint.2020.103237

[*] Corresponding author.
E-mail addresses: mthrun@mathematik.uni-marburg.de (M.C. Thrun), ultsch@mathematik.uni-marburg.de (A. Ultsch).

https://doi.org/10.1016/j.mex.2020.101093

Specifications table

Subject Area	• *Computer Science*
More specific subject area:	*Unsupervised Machine Learning*
Method name:	*Topographic Map Generation Using the Generalized Umatrix*
Name and reference of original method	*Thrun, M. C., & Ultsch, A.: Swarm Intelligence for Self-Organized Clustering, Journal of Artificial Intelligence*, in press, doi:10.1016/j.artint.2020.103237, *2020.* *Ultsch, A., Thrun, M.: Credible Visualizations for Planar Projections, Proc. Workshop on Self-Organizing Maps (WSOM), pp. 256–260, Nancy, 2017.* *Thrun, M. C., Lerch, F., Lötsch, J., Ultsch, A.: Visualization and 3D Printing of Multivariate Data of Biomarkers, In: Skala, V. (Ed.): 24th Conf. on Computer Graphics, Visualization and Computer Vision, Plzen, pp.* 7–16 *2016.* *Ultsch, A.: Maps for the Visualization of high-dimensional Data Spaces, Proc. Workshop on Self organizing Maps (WSOM), pp. 225–230, Kyushu, Japan, 2003.* *Ultsch, A., Siemon, H.P.: Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, In Proceedings Intern. Neural Networks, Kluwer Academic Press, Paris, pp. 305–308, 1990.*
Resource availability	https://CRAN.R-project.org/package=GeneralizedUmatrix

This article is divided into four parts. The first describes the unsupervised artificial network of self-organizing maps (SOMs) in general; the second part describes the application of simplified emergent self-organizing maps (sESOM) for projection methods and the third part describes the visualization of the topographic map with hypsometric tints based on the output of sESOM. The last part shows the application in three examples. The method is part of [30]; it is a co-submission of that article (ARTINT_103237) and the method's description originates from several sections of the Ph.D. thesis, “Projection-Based Clustering through Self-Organization and Swarm Intelligence” [27].

Emergent self-organizing map (ESOM)

Self-organizing (feature) map (SOM) was invented by [13],[14] and is a type of unsupervised neural learning algorithm. In contrast to other neural network models[1] a SOM consists of an ordered two-dimensional layer of neurons called units. Neurons are interconnected nerve cells in the human neocortex [H. [25], p. 22], and the SOM approach was inspired by somatosensory maps (e.g. see [[9], p. 421] cites [8], see also [[12], p. 335]). There are two types of SOM algorithms: online and batch [7]. The first is stochastic, whereas the second is deterministic, which means that it yields reproducible results for a given parameter setting. However, Fort et al. have argued “that randomness could lead to better performances” [[7], p. 12].

The main differences between batch-SOM [16] and online-SOM [15] lie in the updating and averaging of the input data. In batch-SOM, prototypes (see Eq. (1) below) are assigned to the data points and the influences of all associated data points are calculated simultaneously, in contrast to online-SOM, in which sequential training of the neurons is applied (as described in detail below). The batch-SOM method has been shown to produce topographic mappings of varying quality depending on the pre-defined parametrization [7], and “the representation of clusters in the data space on maps trained with batch learning is poor compared to sequential training“ [21]. An important comparison between the batch-SOM approach and ant-based clustering was presented by [10] and will be elaborated upon in chapter 7. No objective function is used in online-SOM [[17], p. 241], and SOM remains a reference tool for two-dimensional visualization [[17], p. 244].

In one common approach to applying the SOM concept, the algorithm acts as an extension of the k-means algorithm [4] or is a partitioning method of the k-means type [20]. In such a case, only a few units are used in the SOM algorithm to represent the data [23], which results in direct clustering of the data. Here, each neuron can be considered to represent a cluster. For example, Cottrell and de

[1] For an overview, see [H. [25]], for deep learning see [Goodfellow et al., 2016].

Bodt used 4 × 4 units to represent the 150 data points in the Iris dataset ([Ultsch et al., 2016a] cites [3]). Therefore, the conventional SOM algorithm is called k-means-SOM here. This SOM algorithm also has two common extensions called Heskes-SOM [11] and Cheng-SOM; these two extensions include objective functions [1] and are not discussed further in this thesis. The optimization of objective functions in general will be discussed in chapter 6, where it will be argued that it is not useful for the goal of this thesis. Chapter 7 will show that objective functions are incompatible with self-organization.

The other approach to applying SOM is to exploit its emergent phenomena through self-organization, in which case it is necessary to use a large number of neurons (>4000) [31]. This enhancement of the online-SOM approach is called emergent SOM (ESOM). In such a case, the neurons serve as a projection of the high-dimensional input space instead of a clustering, as is the case in k-means-SOM.

Let $M = \{m_1, \ldots, m_n\}$ be the positions of n neurons on a two dimensional lattice[2] (feature map) and $W = \{w(m_i) = w_i \mid i = 1, \ldots n\}$ the corresponding set of weights or prototypes of n neurons, then, the SOM training algorithm constructs a non-linear and topology-preserving mapping of the input space I by finding the best matching unit (bmu) for each $l \in I$:

$$bmu(l) = \underset{m_i \in M}{\operatorname{argmin}} \{D(l, w_i)\}, \ \ i \in \{1, \ldots, n\} \tag{1}$$

if in Eq. (1) a distance in the input space I between the point l and the prototype w_i is denoted.

In each step, SOM learning is achieved by modifying the prototypes (weights) in a neighborhood in Eq. (2).

$$\Delta w(R) = \eta(R) * h(bmu(l), m_i, R) * (l - w(m_i)) \tag{2}$$

The cooling scheme is defined by the neighborhood function $h : M \times M \times \mathbb{R}^+ \to [-1, 1]$ and the learning rate η: $\mathbb{R}^+ \to [0, 1]$, where the radius R decreases until $R = 1$ in accordance with the definition of the maximum number of epochs. In contrast to all previously introduced projection methods, no objective function is used in the ESOM algorithm. Instead, ESOM uses the concept of self-organization (see chapter 6 for further details) to find the underlying structures in data.

The structure of a (feature) map is **toroidal**; i.e., the borders of the map are cyclically connected [31], which allows the problem of neurons on borders and, consequently, boundary effects to be avoided. The positions $m \in M$ of the BMUs exhibit no structure in the input space [31]. The structure of the input data emerges only when a SOM visualization technique called U-matrix is exploited [34].

Let $N(j)$ be the eight immediate neighbors of $m_j \in M$, let $w_j \in W$ be the corresponding prototype to m_j, then the average of all distances between prototypes w_i

$$u(j) = \frac{1}{n} \sum_{i \in N(j)} D\big(w(m_i), w\big(m_j\big)\big), \ \ n = |N(j)| \tag{3}$$

A display of all U-heights in Eq. (3) is called a U-matrix [34].

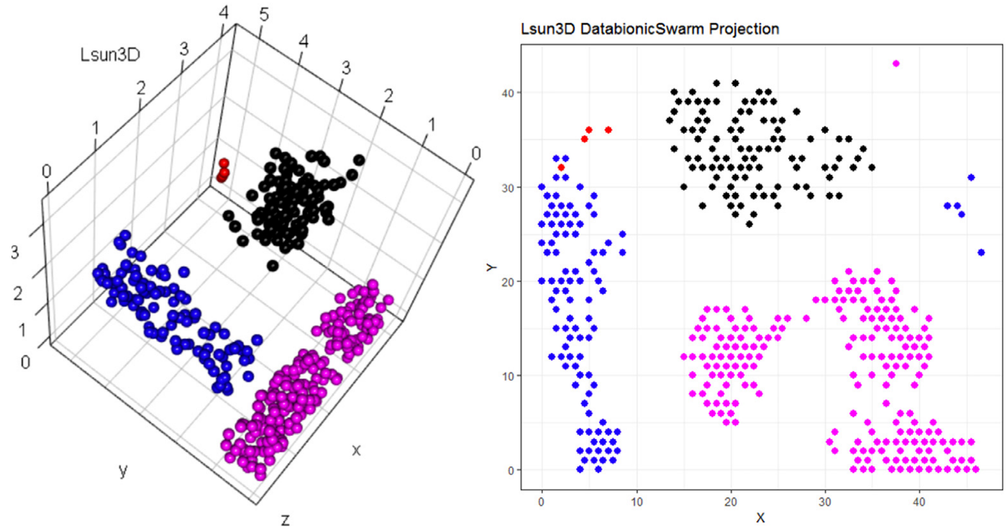
The U-matrix technique that is generally applicable for all projection methods and can be used to visualize both distance- and density-based structures [27],[35]. This visualization technique is the further development of the idea that the U-matrix can be applied to every projection method [33].

In this work, the visualization technique results in a topographic 3D landscape. Here, the requirements are a heavily modified emergent self-organizing map (ESOM) algorithm and a method of high-dimensional density estimation. Contrary to [33], the process of computing the resulting topographic map is completely free of parameter dependence and accessible by simply by downloading the corresponding R package [Thrun/Ultsch, 2017b].

Simplified ESOM

To calculate a U*-matrix for any projection method, a modified ESOM algorithm is required. The first step is the computation of the correct lattice size.

[2] In general this work uses the term grid if the resulting tiling is hexagonal and lattice if the resulting tiling is rectangular (see connected graph). In the context here the distinction is not important, therefore we use the term (feature) map.



Listing 1. Lsun3D data set and DBS projection [Thrun/Ultsch, 2020]. The projection does not indicate which projected points are similar to each other with regards to the high-dimensional structures. The projected points at the top are in Figure 2 at the bottom on the projected points on the left are in Figure 2 on the right.

On the x axis, let the lattice begin at 1 and end at a maximal number denoted by Columns C (equal to the number of columns in the lattice); similarly, on the y axis, let the lattice begin at a maximal number denoted by Lines L and end at 1. Then, the first condition is expressed as

$$\frac{L-1}{C-1} \approx \frac{|\max(y) - \min(y)|}{|\max(x) - \min(x)|} = \frac{dy}{dy} = \Delta \tag{I}$$

The second condition is that the lattice size should be larger than NN[3]:

$$L{*}C \geq \mathrm{NN} \tag{II}$$

The first condition (I.) implies that the lattice size should be as close to equal to the size of the coordinate system as possible. The second condition (II.) is required for emergence in our algorithm (for details, see [31]). The resulting equation to be solved is Eq. (4)

$$L^2 + L(1+\Delta) - \mathrm{NN} * \Delta \geq 0 \tag{4}$$

which yields Eq. 5:

$$L \geq -\frac{1+\Delta}{2} + \sqrt{\left(\frac{1+\Delta}{2}\right)^2 + \mathrm{NN} * \Delta} \tag{5}$$

After the transformation from the projected points[4] $p \in O$ to points on a discrete lattice, the points are called the best-matching units (BMUs) $bmu \in B \subset \mathbb{R}^2$ of the high-dimensional data points j, analogous to the case for general SOM algorithms with $fgrid: O \to B, p \mapsto bmu$, where *fgrid* is surjective when conditions (i) and (ii) are met.

To develop the algorithm illustrated in Listing 1, the idea of [33], in which it was suggested to "apply Self-Organizing Map training without changing the best match[ing unit] assignment", was adopted. However, in contrast to [33], here, the transformation *fgrid* is defined precisely to calculate

[3] In [31] the minimum number of 4096 neuros was proposed.

[4] Or DataBot positions on the hexagonal grid of Pswarm (see [30]).

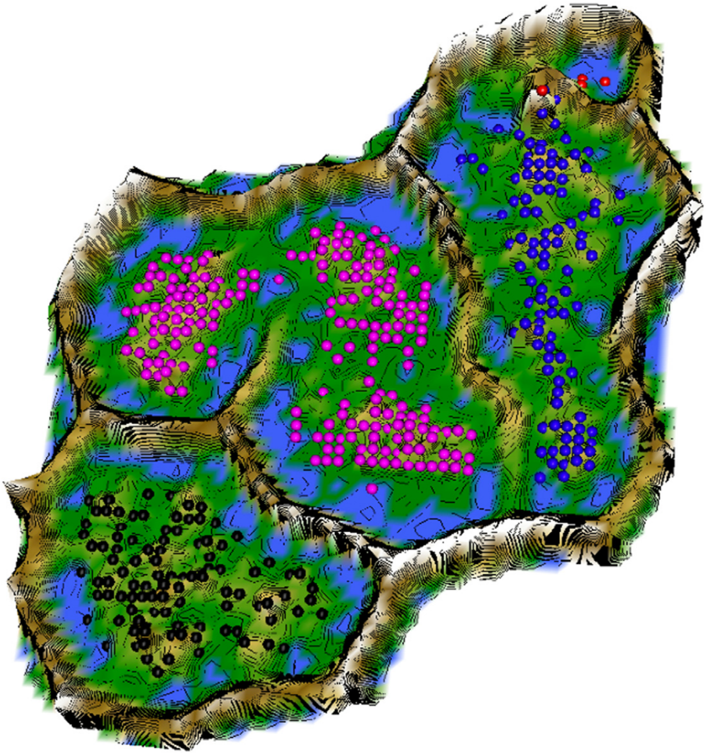


Fig. 1. Topographic map of the projection shows a high-dimensional structures which corresponds to the prior classification of Lsund3D. The four Outliers, labeled in red, lie either on a volcano or a high mountain. The borders of the scatter plot in Figure 1 (right), which divide the blue cluster in two parts, are ignored.

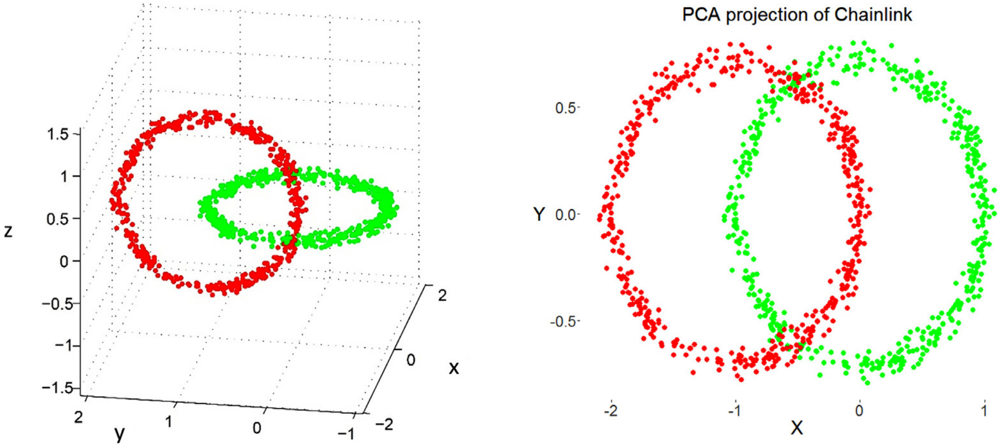


Fig. 2. Chainlink data set (right) and PCA projection. The projection suffers from local errors in two small areas around a low number of points, but the projection is unable to visualize them.

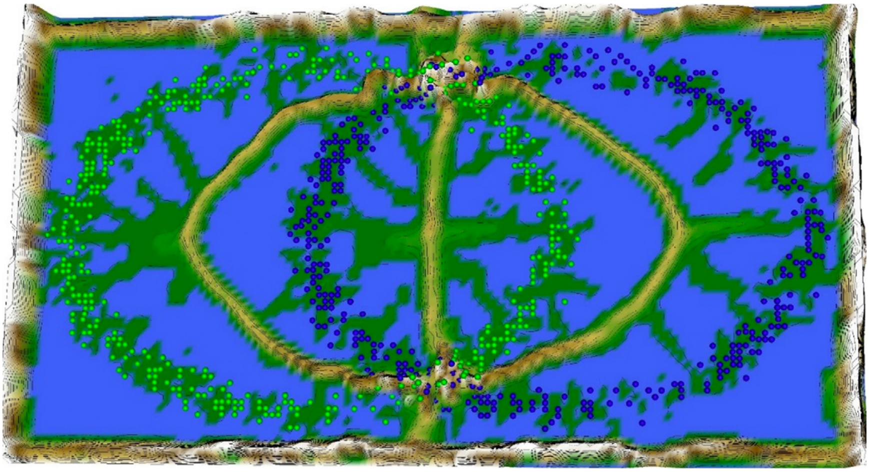


Fig. 3. Topographic map of the PCA projection of the Chainlink data set. The errors of the projection are clearly visible. Contrary to Figure 1 and 2, the borders of the scatter plot are visualized as hills in the topographic map showing that points on the borders are far away from each other.

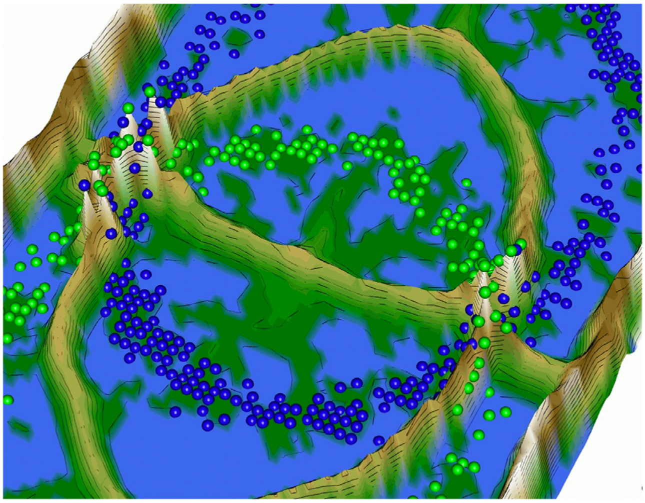


Fig. 4. Zoomed-in view of the misrepresentation of the relative relationship between the high-dimensional points (structures) of the Chainlink dataset in the PCA projection.

the BMU positions and the structure of the lattice is toroidal; i.e., the borders of the lattice are cyclically connected [31].

Based on the relevant symmetry considerations,[5] a simplified version of ESOM (sESOM) is introduced here. No epochs or learning rate are required, because the cooling scheme is defined by a special neighborhood function $\mathrm{h} : M \times M \times \mathbb{R}^{+} \to [0, 1]$.

Let $M = \{m_1, \ldots, m_n\}$ be a set of neurons (where m_i are the lattice positions) with the corresponding prototype set $W = \{w_1, \ldots, w_n\}$, where dim(W)=dim(I) and $\#W$=#M; then, the

[5] See [30] for details.

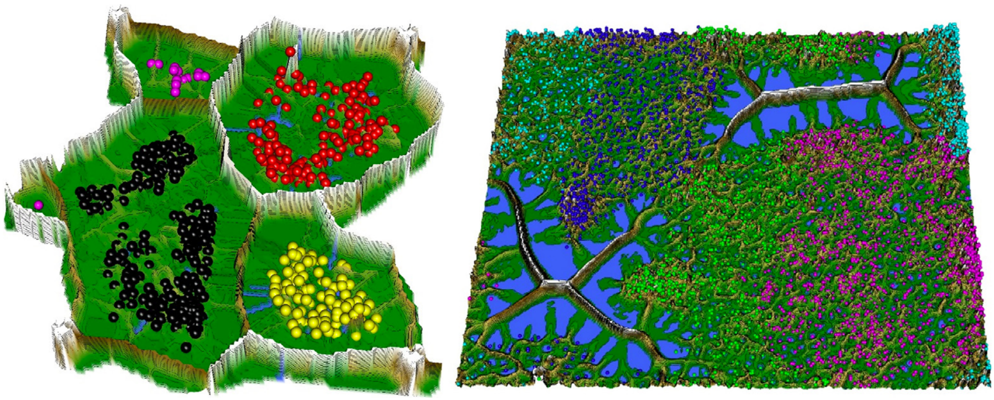


Fig. 5. Left: topographic map of the normalized generalized U-matrix using the DBS projection module [Thrun/Ultsch, 2020] visualizes the distance-based structures of 7447 dimensions. Each point represents a patient colored by their diagnosis. Patients with the same diagnosis lie in the same valley. Right: topographic map of generalized U-matrix using the DBS projection module [Thrun/Ultsch, 2020] visualizes the absence of distance-based structures for the Golfball dataset (see Data in Brief article). No valleys are visible and colored points are not separated by watersheds of hills and mountain ranges.

neighborhood function h is defined in Eq. (6):

$$h = \begin{cases} 1 - \frac{\mathrm{d(j,l)}^2}{\pi \mathrm{R}^2}, \; iff \; \frac{\mathrm{d(j,l)}^2}{\pi \mathrm{R}^2} < 1 \\ 0, \; else \end{cases} \tag{6}$$

In sESOM, learning is achieved in each step by modifying the weights in a neighborhood in Eq. (7):

$$\Delta w(R) = 1 * h(bmu(j), m_i, R) * (j - w(m_i)) \tag{7}$$

In contrast to [33], the algorithm does not require any input parameters, and the resulting visualization is not a two-dimensional gray-scale map but rather a topographic map with hypsometric tints [28]. The entire algorithm is summarized in Listing 1.

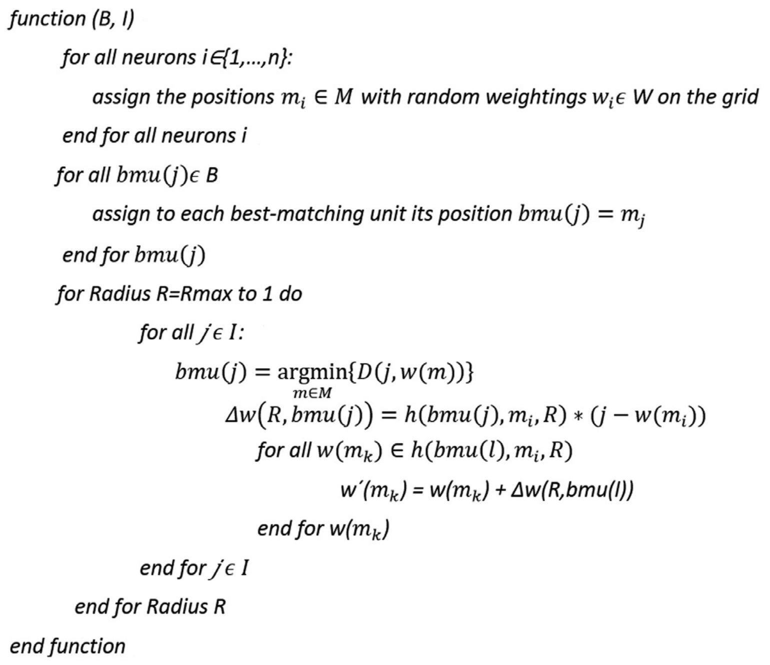
Topographic map with hypsometric

The U*-matrix visualization technique produces a topographic map with hypsometric tints [28]. Hypsometric tints are surface colors that represent ranges of elevation [22]. Here, a specific color scale is combined with contour lines.

The color scale is chosen to display various valleys, ridges and basins: blue colors indicate small distances (sea level), green and brown colors indicate middle distances (low hills), and shades of white colors indicate large distances (high mountains covered with snow and ice). Valleys and basins represent clusters, and the watersheds of hills and mountains represent the borders between clusters (Fig. 2 and Fig. 6).

The landscape consists of receptive fields, which correspond to certain U*-height intervals with edges delineated by contours. This work proposes the following approach (see [[28], p. 10]): First, the range of U*-heights is split up into intervals, which are assigned uniformly and continuously to the color scale described above through robust normalization [18]. In the next step, the color scale is interpolated based on the corresponding CIELab color space [2]. The largest possible contiguous areas corresponding to receptive fields in the same U*-height intervals are outlined in black to form contours. Consequently, a receptive field corresponds to one color displayed in one particular location in the U*-matrix visualization within a height-dependent contour. Let u(j) denote the U*-heights, and let q01 and q99 denote the first and 99-th percentiles, respectively, of the U*-heights; then, the robust normalization of the U*-heights u(j) is defined by Eq. (8):

$$u(j) = \frac{u(j) - q01}{q99 - q01} \tag{8}$$



```
function (B, I)
    for all neurons i∈{1,...,n}:
        assign the positions $m_i \in M$ with random weightings $w_i \epsilon$ W on the grid
    end for all neurons i
    for all $bmu(j) \epsilon$ B
        assign to each best-matching unit its position $bmu(j) = m_j$
    end for $bmu(j)$
    for Radius R=Rmax to 1 do
            for all $j \epsilon I$:
                $bmu(j) = \underset{m \in M}{\operatorname{argmin}}\{D(j, w(m))\}$
                    $\Delta w(R, bmu(j)) = h(bmu(j), m_i, R) * (j - w(m_i))$
                        for all $w(m_k) \in h(bmu(l), m_i, R)$
                                $w'(m_k) = w(m_k) + \Delta w(R,bmu(l))$
                        end for $w(m_k)$
            end for $j \epsilon I$
      end for Radius R
end function
```

Fig. 6. Listing 1: sESOM pseudocode algorithm implements a stepwise iteration from the maximum radius Rmax which is given by the lattice size (Rmax = C/6) stepwise with one per step and down to 1. w(́m_k) indicates that the prototype w(m_k) of neuron m_k is modified by Eq. 7 Additionally, the search for a new best matching unit still is used and these prototypes may change during one iteration. The predefined prototypes are reset to the weights of their corresponding high-dimensional data points after each iteration.

The number of intervals in is defined by Eq. (9):

$$\frac{1}{in} = \frac{q01}{q99} \tag{9}$$

The resulting visualization consists of a hierarchy of areas of different height levels represented by corresponding colors (see Figure). To the human eye, the visualization using the generalized U-matrix tool is analogous to a topographic map; therefore, one can visually interpret the presented data structures in an intuitive manner. In contrast to other SOM visualizations, e.g., [26], this topographic map presentation enables the layman to interpret sESOM results.

The use of a toroidal map for sESOM computations necessitates a tiled landscape display in the interactive generalized U-matrix tool [29], which means that every receptive field is shown four times. Consequently, in the first step, the visualization consists of four adjoining images of the same generalized U-matrix [32] (the same is true for the U*-matrix). To obtain the 3D landscape (island[6]), a shiny application can be used to cut an island out.

Summary

Restricting the Output space of a Projection method results in projection errors because the two-dimensional similarities in the scatter plot cannot coercively represent high-dimensional distances. This is stated by the Johnson–Lindenstrauss lemma [5] and visualized in two examples. Fig. 1 and Fig. 3 show, how scatter plots lead to misleading interpretation of the high-dimensional structures.

[6] An island can be also cut interactively (or the cutting may be improved) and thus may not be rectangular

However, scatter plots of projection methods remain the state of the art in cluster analysis as a visualization of distance and density-based structures (e.g., [[6], pp. 31–32; [19], p. 25; G. [24], p. 223; [9], pp. 119–120, 683–684]). Thus, the projected points of such a scatter plot are used in a simplified emergent self-organizing map in order to compute a generalized U-matrix. This generalized U-matrix defines the visualization of a topographic map which provides more accurate information about the high-dimensional distance and density based structures of the data. The topographic maps of the three examples are visualized in Figs. 1–6. For Fig. 6 the projections are not shown.

Example 1: Linear Separable Structures
Example 2: Linear Non-Separable Structures
Example 3: High-Dimensional Structures versus the Absence of Such Structures

Declaration of Competing Interest

None.

References

[1] Y. Cheng, Convergence and ordering of Kohonen's batch map, Neural Comput. 9 (8) (1997) 1667–1676.
[2] Colorimetry C.I.E, Vol., CIE Publication,Central Bureau of the CIE, Vienna, 2004.
[3] Marie Cottrell, Eric de Bodz, A Kohonen map representation to avoid misleading interpretations, in: ESANN, 96, Ciaco, Bruges, Belgium, 1996.
[4] M. Cottrell, M. Olteanu, F. Rossi, N. Villa-Vialaneix, Theoretical and applied aspects of the self-organizing maps, in: Advances in Self-Organizing Maps and Learning Vector Quantization, Springer, 2016, pp. 3–26.
[5] S. Dasgupta, A. Gupta, An elementary proof of a theorem of Johnson and Lindenstrauss, Random Struct. Algorithms 22 (1) (2003) 60–65.
[6] B.S. Everitt, S. Landau, M. Leese, L. McAllister (Ed.), Arnold, London, 2001 ISBN: 978-0-340-76119-9.
[7] J.-.C. Fort, M. Cottrell, P. Letremy, Stochastic on-line algorithm versus batch algorithm for quantization and self organizing maps, in: Proceeding of the Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop, IEEE, 2001, pp. 43–52.
[8] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, Uppder Saddle River, NJ, 1994 ISBN: 0023527617.
[9] C. Hennig, et al., (Hg.): *Handbook of Cluster Analysis*, Chapman&Hall/CRC Press, New York, USA, 2015 ISBN: 9781466551893.
[10] L. Herrmann, A. Ultsch, Explaining ant-based clustering on the basis of self-organizing maps, Proc. ESANN (2008) 215–220 Citeseer.
[11] T. Heskes, Energy functions for self-organizing maps, Kohonen Maps (1999) 303–316.
[12] E. Kandel, Neurowissenschaften. Eine Einführung, Padiatrische Praxis 79 (4) (2012) 672.
[13] T. Kohonen, Analysis of a simple self-organizing process, Biol. Cybern. 44 (2) (1982) 135–140.
[14] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. cybern. 43 (1) (1982) 59–69.
[15] T. Kohonen, Self-Organizing Maps, 30, Springer, Berlin Heidelberg, 1995 ISBN: 978-3-642-97610-0.
[16] T. Kohonen, P. Somervuo, How to make large self-organizing maps for nonvectorial data, Neural Netw. 15 (8) (2002) 945–952.
[17] J.A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, Springer, New York, USA, 2007 ISBN: 978-0-387-39350-6.
[18] G.W. Milligan, M.C. Cooper, A study of standardization of variables in cluster analysis, J. Classificat. 5 (2) (1988) 181–204.
[19] B.G. Mirkin, Clustering: a Data Recovery Approach, Chapnman&Hall/CRC, Boca Raton, FL, USA, 2005 ISBN: 978-1-58488-534-4.
[20] F. Murtagh, M. Hernández-Pajares, The Kohonen self-organizing map method: an assessment, J. Classificat. 12 (2) (1995) 165–190.
[21] M. Nöcker, F. Mörchen, A. Ultsch, An algorithm for fast and reliable ESOM learning, Proc. ESANN (2006) 131–136.
[22] T. Patterson, N.V. Kelso, Hal Shelton revisited: designing and producing natural-color maps with satellite land cover data, Cartogr. Perspect. 47 (2004) 28–55.
[23] T. Reutterer, Competitive market structure and segmentation analysis with self-organizing feature maps, in: Proc. Proceedings of the 27th EMAC Conference, Citeseer, 1998, pp. 85–115.
[24] G. Ritter, Robust Cluster Analysis and Variable Selection, Chapman&Hall/CRC Press, Passau, Germany, 2014 ISBN: 1439857962.
[25] H. Ritter, T. Martinetz, K. Schulten, D. Barsky, M. Tesch, R. Kates, Neural Computation and Self-Organizing Maps: an Introduction, Addison-Wesley Reading, MA, 1992 ISBN: 0201554429.
[26] K. Tasdemir, E. Merenyi, Exploiting data topology in visualization and clustering of self-organizing maps, IEEE Trans. Neural Netw. 20 (4) (2009) 549–562, doi:10.1109/tnn.2008.2005409.
[27] M.C. Thrun, A. Ultsch, E. Hüllermeier (Eds.), Springer, Heidelberg, 2018 Doctoral dissertation, doi:10.1007/978-3-658-20540-9ISBN: 978-3658205393.
[28] M.C. Thrun, F. Lerch, J. Lötsch, A. Ultsch, Visualization and 3D printing of multivariate data of biomarkers, in: V. Skala (Ed.), International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), 24, 2016, pp. 7–16. http://wscg.zcu.cz/wscg2016/short/A43-full.pdf. Plzen.
[29] M.C. Thrun, F. Pape, A. Ultsch, Interactive Machine Learning Tool for Clustering in Visual Analytics, in: 7th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2020), Sydney, Australia, 2020, pp. 672–680, doi:10.1109/DSAA49011.2020.00062.

[30] M.C. Thrun, A. Ultsch, Swarm intelligence for self-organized clustering, J. Artif. Intell. (2020) In press, doi:10.1016/j.artint.2020.103237.

[31] A. Ultsch, Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series, in: E. Oja, S. Kaski (Eds.), Kohonen Maps, 1 ed., Elsevier, 1999, pp. 33–46.

[32] A. Ultsch, Maps for the visualization of high-dimensional data spaces, in: Proceeding of the Workshop on Self organizing Maps (WSOM), Kyushu, Japan, 2003, pp. 225–230.

[33] A. Ultsch, F. Mörchen, U-maps: topograpic visualization techniques for projections of high dimensional data, in: Proc. Proc. 29th Annual Conference of the German Classification Society, Citeseer, 2006.

[34] A. Ultsch, H.P. Siemon, Kohonen's self organizing feature maps for exploratory data analysis, in: International Neural Network Conference, Paris, France, Kluwer Academic Press, 1990, pp. 305–308.

[35] A. Ultsch, M.C. Thrun, Credible visualizations for planar projections, in: M. Cottrell (Ed.), 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM), IEEE, Nany, France, 2017, pp. 1–5, doi:10.1109/WSOM.2017.8020010.