Self-Organizing Map Convergence

Robert Tatoian **Lutz Hamel**
Department of Computer Science and Statistics
University of Rhode Island
Kingston, Rhode Island, USA
hamel@cs.uri.edu

Abstract

Self-organizing maps are artificial neural networks designed for unsupervised machine learning. They represent powerful data analysis tools applied in many different areas including areas such as biomedicine, bioinformatics, proteomics, and astrophysics. We maintain a data analysis package in R based on self-organizing maps. The package supports efficient, statistical measures that enable the user to gauge the quality of a generated map. Here we introduce a new quality measure called the convergence index. The convergence index is a linear combination of map embedding accuracy and estimated topographic accuracy and since it reports a single statistically meaningful number it is perhaps more intuitive to use than other quality measures. Here we study the convergence index in the context of clustering problems proposed by Ultsch as part of his fundamental clustering problem suite. We demonstrate that the convergence index captures the notion that a SOM has learned the multivariate distribution of a training data set.

1 Introduction

Self-organizing maps are artificial neural networks designed for unsupervised machine learning. They represent powerful data analysis tools applied in many different areas including areas such as biomedicine, bioinformatics, proteomics, and astrophysics [1]. We maintain a data analysis package in R called `popsom` [2] based on self-organizing maps. The package supports efficient, statistical measures that enable the user to gauge the quality of a generated map [3]. Here we introduce a new quality measure called the *convergence index*. The convergence index is a linear combination of map embedding accuracy and estimated topographic accuracy. It reports a single, statistically meaningful number between 0 and 1 – 0 representing a least fitted model and 1 representing a completely fitted model – and is therefore perhaps more intuitive to use than other quality measures. Here we study the convergence index in the context of clustering problems proposed by Ultsch as part of his fundamental clustering problem suite [4]. In particular, we are interested in how well the convergence index captures the notion that a SOM has learned the multivariate distribution of a training data set.

Over the years a number of different quality measures for self-organizing maps have been proposed. Nice overviews of common SOM quality measures appear in [5] and [6]. Our convergence index distinguishes itself from many of the other measures in that it is statistical in nature. This is particularly true for the part of the convergence index based on embedding (or coverage) which is essentially a two-sample test between the training data and the set of self-organizing map neurons viewed as populations. The two sample test measures how similar these two populations are. For a fully embedded map the population of neurons should be indistinguishable from the training data. This statistical view of embedding is interesting because it makes the standard visualization of SOMs using a U-matrix [7] statistically meaningful. That is, the cluster and the distance interpretations of the U-matrix now have a statistical foundation based on the fact that the distribution of the map neurons is indistinguishable from the distribution of the training data.

The other part of our convergence index, the estimated topographic accuracy, is an efficient statistical approach to the usual topographic error quality measure [5] which can be computationally expensive. In our approach we use a sample of the training data to estimate the topographic accuracy. Experiments have shown that we need only a fairly small sample of the training data to get very accurate estimates.

The remainder of this paper is structured as follows. In Section 2 we briefly review self-organizing maps as implemented by our package. We take a look at the quality measures implemented in `popsom` in Section 3. In Section 4 we define our convergence index. We discuss our case studies in Section 5. Conclusions and further work are discussed in Section 6

2 Self-Organizing Maps

Briefly, a self-organizing map [1] is a kind of artificial neural network that implements competitive learning, which can be considered a form of unsupervised learning. On the map itself, neurons are arranged along a rectangular grid with dimensions x_{dim} and y_{dim}. Learning proceeds in two steps for each training instance $\vec{x}_k$, $k = 1, 2, 3, \ldots, M$, with M the number of training instances:

1. The **competitive step** where the best matching neuron for a particular training instance is found on the rectangular grid,

$$c = \operatorname*{argmin}_i(||\vec{m}_i - \vec{x}_k||)$$

where $i = 1, 2, \ldots, N$ is an index over the neurons of the map with $N = x_{dim} \times y_{dim}$ is the number of neurons on the grid, and $\vec{m}_i$ is a neuron indexed by i. Finally, c is the index of the best matching neuron $\vec{m}_c$ on the map.

2. The **update step** where the training instance $\vec{x}_k$ influences the best matching neuron $\vec{m}_c$ and its neighborhood. The update step can be represented by the following update rule for the neurons on the map,

$$\vec{m}_i \leftarrow \vec{m}_i - \eta\vec{\delta}_i h(c, i)$$

for $i = 1, 2, \ldots, N$. Here $\vec{\delta}_i = \vec{m}_i - \vec{x}_k$, η is the learning rate, and $h(c, i)$ is a loss function with,

$$h(c, i) = \begin{cases} 1 & \text{if } i \in \Gamma(c), \\ 0 & \text{otherwise}, \end{cases}$$

where $\Gamma(c)$ is the neighborhood of the best matching neuron $\vec{m}_c$ with $c \in \Gamma(c)$. Typically the neighborhood is a function of time and its size decays during training. Initially the neighborhood for neuron $\vec{m}_c$ includes all other neurons on the map,

$$\Gamma(c)|_{t=0} = \{1, 2, \ldots, N\}.$$

As training proceeds the neighborhood for $\vec{m}_c$ shrinks down to just the neuron itself,

$$\Gamma(c)|_{t\gg 0} = \{c\}.$$

Here, as before, $N = x_{dim} \times y_{dim}$ is the number of neurons on the map. This means that initially the update rule for each best matching neuron has a very large field of influence which gradually shrinks to the point that the field of influence just includes the best matching neuron itself.

The two training steps above are repeated for each training instance until the given map converges.

Figure 1 shows a scatter plot of the Tetra problem in Ultsch's fundamental clustering problem suite (FCPS) [4]. The data set consists of four almost touching clusters embedded in three dimensional space. Figure 2 shows a SOM starburst plot of this data set generated with our `popsom` package [2] which supports statistical convergence criteria [3] and detailed cluster visualizations in terms of our starburst plots [8].

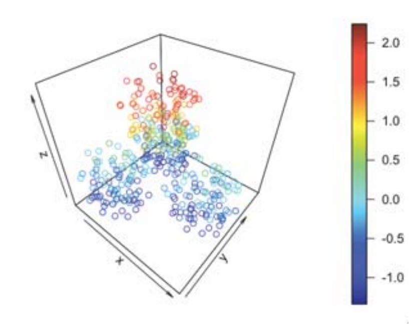


Figure 1: The Tetra data set.

The four clusters can easily be identified on the map by their starbursts. Also easily visible is the fact that clusters themselves are identified by their light color and cluster boundaries are identified by darker colors. The easily identified borders mean that the clusters are indeed distinct clusters. Their relative position is also meaningful to a point, given that this is a 2D rendering of a higher dimensional space. All these observations are justified due to the fact that the map has converged and therefore positioning and distance amongst clusters is statistically meaningful.

3 Quality Measures

3.1 Map Embedding Accuracy

Yin and Allinson have shown that under some mild assumptions the neurons of a large enough self-organizing map will converge on the probability distribution of the training data given infinite time [9]. This is the motivation for our map embedding accuracy:

> *A SOM is completely embedded if its neurons appear to be drawn from the same distribution as the training instances.*

This view of embedding naturally leads to a two-sample test [10]. Here we view the training data as one sample from some probability space $\mathbf{X}$ having the probability density function $p(x)$ and we treat the neurons of the SOM as another sample. We then test to see whether or not the two samples appear to be drawn from the same probability space. If we operate under the simplifying assumption that each of the d features of the input space $\mathbf{X} \subset \mathbb{R}^d$ are normally distributed and independent of each other, we can test each of the features separately. This assumption leads to a fast algorithm for identifying SOM embedding: We define a feature as embedded if the variance and the mean of that feature appear to be drawn from the same distribution for both the training data and the

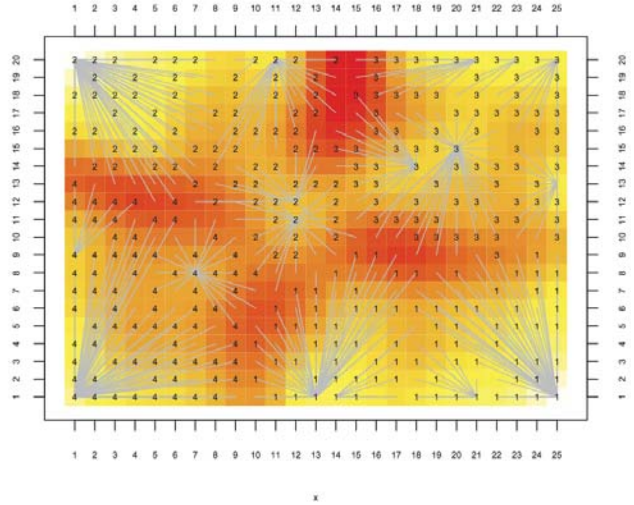


Figure 2: A SOM starburst plot of the Tetra data set.

neurons. If all the features are embedded then we say that the map is *fully embedded*.

The following is the formula for the $(1-\alpha)*100\%$ confidence interval for the ratio of the variances from two random samples [10],

$$\frac{s_1^2}{s_2^2} \cdot \frac{1}{f_{\frac{\alpha}{2},n_1-1,n_2-1}} < \frac{\sigma_1^2}{\sigma_2^2} < \frac{s_1^2}{s_2^2} \cdot f_{\frac{\alpha}{2},n_1-1,n_2-1}, \qquad (1)$$

where s_1^2 and s_2^2 are the values of the variance from two random samples of sizes n_1 and n_2 respectively, and where $f_{\frac{\alpha}{2},n_1-1,n_2-1}$ is an F distribution with n_1-1 and n_2-1 degrees of freedom. To test for SOM embedding, we let s_1^2 be the variance of a feature in the training data and we let s_2^2 be the variance of that feature in the neurons of the map. Furthermore, n_1 is the number of training samples and n_2 is the number of neurons in the SOM. The variance of a particular feature of both training data and neurons appears to be drawn from the same probability space if 1 lies in the confidence interval denoted by equation (1): the ratio of the underlying variance as modeled by input space and the neuron space, respectively, is approximately equal to one, $\sigma_1^2/\sigma_2^2 \approx 1$, up to the confidence interval.

In the case where $\bar{x}_1$ and $\bar{x}_2$ are the values of the means from two random samples of size n_1 and n_2, and the variances of these samples are σ_1^2 and σ_2^2 respectively, the following formula provides $(1-\alpha)*100\%$ confidence interval for the difference between the means [10],

$$\mu_1 - \mu_2 \quad > \quad (\bar{x}_1 - \bar{x}_2) - z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}, \qquad (2)$$

$$\mu_1 - \mu_2 \quad < \quad (\bar{x}_1 - \bar{x}_2) + z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}. \qquad (3)$$

The mean of a particular feature for both training data and neurons appears to be drawn from the same probability space if 0 lies in the confidence interval denoted by equations (2) and (3). Here $z_{\frac{\alpha}{2}}$ is the appropriate z score for the chosen confidence interval.

We say that a feature is embedded if the above criteria for both the mean and variance of that feature are fulfilled. We can now define the *map embedding accuracy* for d features,

$$ea = \frac{1}{d}\sum_{i=1}^{d} \rho_i, \qquad (4)$$

where

$$\rho_i = \begin{cases} 1 & \text{if feature } i \text{ is embedded,} \\ 0 & \text{otherwise.} \end{cases}$$

The map embedding accuracy is the fraction of the number of features which are actually embedded (i.e. those features whose mean and variance were adequately modeled by the neurons in the SOM). With a map embedding accuracy of 1 a map is fully embedded. In order to enhance the map embedding accuracy in our implementation [2], we multiply each embedding term ρ_i by the significance of the corresponding feature i which is a Bayesian estimate of that feature's relative importance [11]. A more in-depth statistical analysis of our map embedding accuracy can be found in [12].

3.2 Estimated Topographic Accuracy

Many different approaches to measuring the topological quality of a map exist, *e.g.* [13, 14]. But perhaps the simplest measure of the topological quality of a map is the *topographic error* [15] defined as:

$$te = \frac{1}{n}\sum_{i=1}^{n} err(\vec{x}_i) \tag{5}$$

with

$$err(\vec{x}_i) = \begin{cases} 1 & \text{if } bmu(\vec{x}_i) \text{ and } 2bmu(\vec{x}_i) \text{ are not neighbors,} \\ 0 & \text{otherwise.} \end{cases}$$

for training data $\{\vec{x}_1, \ldots, \vec{x}_n\}$ where $bmu(\vec{x}_i)$ and $2bmu(\vec{x}_i)$ are the best matching unit and the second-best matching unit for training vector $\vec{x}_i$ on the map, respectively. We define the *topographic accuracy* of a map as,

$$ta = 1 - te. \tag{6}$$

Computing the topographic accuracy can be very expensive, especially for large training data sets and/or maps. One way to ameliorate the situation is to sample the training data and use this sample S to estimate the topographic accuracy. If we let s be the size of the sample then the *estimated topographic accuracy* is,

$$ta' = 1 - \frac{1}{s}\sum_{\vec{x}\in S} err(\vec{x}). \tag{7}$$

We have shown in [3] that we can get accurate values for ta' with very small samples.

In addition to computing the value for the estimated topographic accuracy we use the bootstrap [16] to compute values for an appropriate confidence interval in order to give us further insight into the estimated topographic accuracy in relation to the actual value for the topographic accuracy whose value should fall within the bootstrapped confidence interval.

It is easy to see from (7) that for topologically faithful maps the estimated topographic accuracy should be close to 1. We then say that the map is *fully organized*.

4 The Convergence Index

Recently it has been argued that any SOM quality measure needs to report on both the embedding of a map in the input data space as well as the topological quality of a map [17]. In order to have a simple interpretation of the embedding and the topographic accuracy we introduce the convergence index, cix, which is a linear combination of the two quality measures introduced in the previous section,

$$cix = \frac{1}{2}ea + \frac{1}{2}ta'. \tag{8}$$

Table 1: Training results for the Tetra data set.

$iter$	cix	ea	ta'	(lo-hi)
1	0.01	0.00	0.02	(0.00 - 0.06)
10	0.02	0.00	0.04	(0.00 - 0.10)
100	0.35	0.00	0.70	(0.58 - 0.82)
1000	0.61	0.33	0.88	(0.80 - 0.96)
10000	0.99	1.00	0.98	(0.94 - 1.00)
100000	1.00	1.00	1.00	(1.00 - 1.00)

The convergence index is equal to 1 for a fully embedded and fully organized map. In our previous studies [3, 12] we have shown that the embedding accuracy and the estimated topographic accuracy are conservative measures and therefore subsume many of the other SOM quality measures.

5 Case Studies

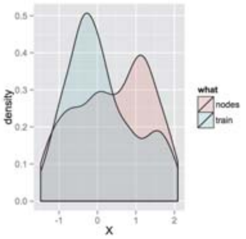
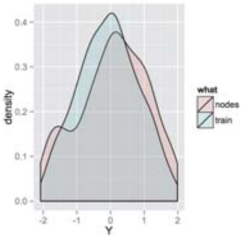
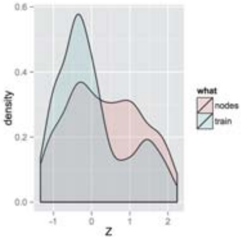
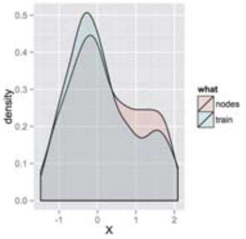
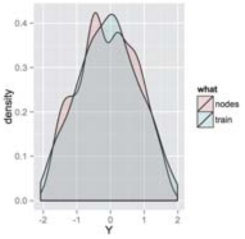
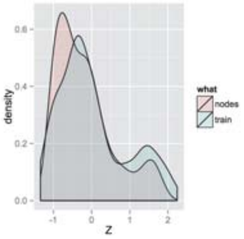
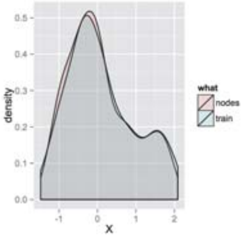
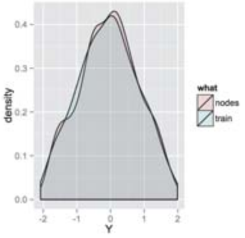
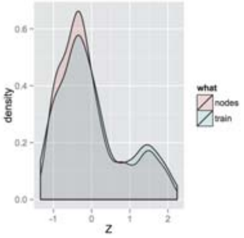
We apply our SOM algorithm to two data sets from Ultsch's fundamental clustering problem suite (FCPS) [4]. We are particularly interested in how the distribution of the neurons converges on the distribution of the training data set as training of the map progresses and how this relates to our convergence index. Our experiments seem to confirm that when $cix \approx 1$ then the distribution of the neurons matches the distribution of the training data almost exactly as measured on the marginals.

5.1 The Tetra Data Set

Our first data set is the Tetra data set. As with all the data sets in FCPS this data set is a synthetic data set with four almost touching clusters in three dimensions. Figure 1 shows a scatter plot of this data set. The four clusters are clearly identifiable. We trained a 25×20 SOM using this data set stepping the training iterations in powers of 10 from 1 to 100000. Table 1 shows the results. We can observe that the convergence index (cix) starts from just about 0 and grows to 1. Also included in the table are the embedding accuracy (ea) and the estimated topographic accuracy (ta') together with its 95% confidence interval (lo-hi).

In Table 2 we show the distribution of the neurons (pink) compared to the distribution of the training data (green) in relation to the cix and the number of training iterations $iter$. We show the distributions of the three marginals X, Y, and Z at training iterations 10, 1000, and 100000, respectively. It is striking to see how precisely the neurons model the training data distribution when $cix = 1$. At least in the case of the X and Y marginals. In the Z marginal we can see some discrepancy and we are wondering if that is due to the fact that we are making the simplifying assumption of a normal distribution when testing for embedding accuracy. One of our research

Table 2: Distribution of the neurons and training data (Tetra data set).

iter	cix	X	Y	Z
10	0.01			
1000	0.61			
100000	1.00			

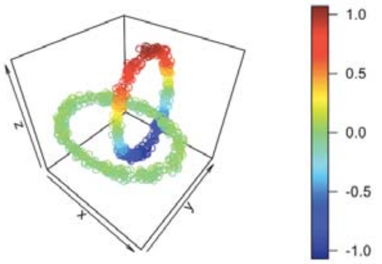


Figure 3: The Chainlink data set.

Table 3: Training results for the Chainlink data set.

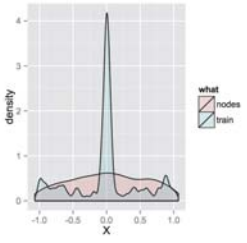
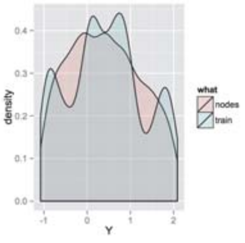
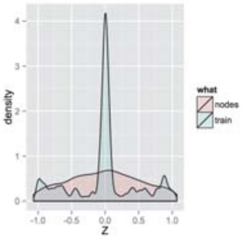
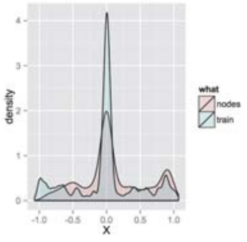
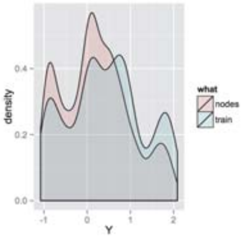
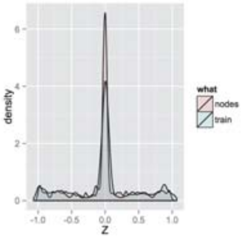
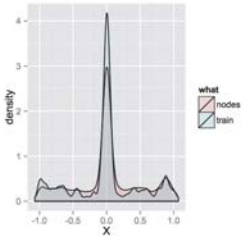
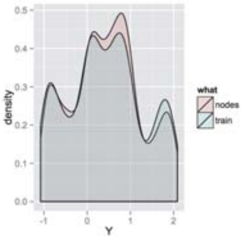
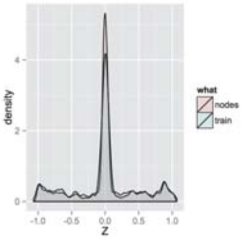
$iter$	cix	ea	ta'	$(lo\text{-}hi)$
1	0.31	0.61	0.00	(0.00 - 0.00)
10	0.33	0.61	0.04	(0.00 - 0.10)
100	0.12	0.00	0.24	(0.14 - 0.38)
1000	0.39	0.00	0.78	(0.66 - 0.90)
10000	0.57	0.19	0.94	(0.84 - 1.00)
100000	0.89	0.81	0.96	(0.90 - 1.00)

goals is to replace the parametric embedding test with a distribution free test. Figure 2 shows the resulting SOM starburst of the Tetra set after 100000 training iterations. As expected, the clusters are well defined and separated given $cix = 1$.

5.2 The Chainlink Data Set

The second data set we look at is the Chainlink data set. The interesting aspect of this data set is that the two classes defined by this data set are not linearly separable. Figure 3 shows a scatter plot of this data set. We proceeded in the same way as we did for the first data set in that we trained a 45×40 SOM in powers of 10 from 1 to 100000. Table 3 shows the results of this training. What is perhaps striking is that the embedding accuracy ea is fairly large initially and then drops off to 0 only to then increase back to something close to 1. A look at Table 4 perhaps explains this in that the randomly generated initial population of neurons happens to model the distribution of the Y marginal quite well even without a lot of training. Notice however that the estimated topographic accuracy for these initial maps is equal to zero. That means this distribution is generated by neurons that are not properly organized on the map. This explains why once

Table 4: Distribution of the neurons and training data (Chainlink data set).

iter	cix	X	Y	Z
10	0.33			
1000	0.39			
100000	0.89			

a larger number of training iterations is applied to the map the embedding accuracy drops down to 0 before properly organized neurons model the distribution. At 100000 iterations we have $cix = 0.89$ and we can see that the distributions on the marginals are modeled quite well. However, for this non-separable data set we do not expect to reach a convergence index of 1 since from a topological point of view it is very to difficult to have a sheet model two interlocking rings.

Figure 4 shows the SOM starburst plot of the Chainlink data set after 100000 iterations. The clusters are well defined and separated with one exception. As suspected, given a convergence index of less than 1, we have a strange fold in the center of the map where the SOM tried to accommodate the interlocking rings. In our experience, even when training is restarted with a different set of initial conditions some sort of anomaly will always emerge while the SOM tries to accommodate the interlocking rings. These anomalies prevent the convergence index to become 1. In some sense this is satisfying, since now we can view the convergence index as a way to also measure the difficulty of the input space.

6 Conclusions and Further Work

Self-organizing maps are powerful data analysis tools applied to many different areas such as biomedicine, genomics, and physics. We maintain a data analysis package in R based on self-organizing maps. The package supports efficient, statistical measures enabling the user to gauge the quality of a given map. Here we introduced a new quality measure called the convergence index and demonstrated that it captures the notion that a SOM has learned the multivariate distribution of a training data set. The advantages of this new quality measure is that it is intuitive and easy to interpret.

Because the SOM algorithm is a constrained learner in the sense that neurons are not able to freely move around the space spanned by the training data it is sufficient to test the marginals for convergence. Our next step is to make this argument statistically rigorous. We also would like to dispense with the normality assumptions in our tests. The one-dimensional Kolmogorov-Smirnov test [18] seems to be suitable here.

References

[1] T. Kohonen, *Self-organizing maps*. Springer Berlin, 2001.

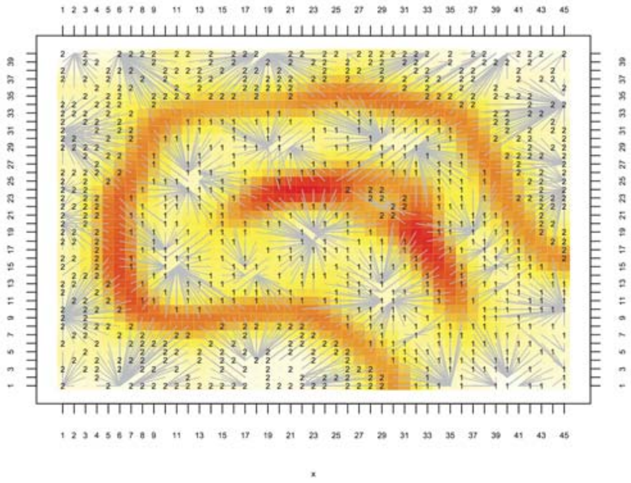


Figure 4: A SOM starburst plot of the Chainlink data set.

[2] L. Hamel, B. Ott, and G. Breard, *popsom: Self-Organizing Maps With Population Based Convergence Criterion*, 2015. R package version 3.0.

[3] L. Hamel, "Som quality measures: An efficient statistical approach," in *Advances in Self-Organizing Maps and Learning Vector Quantization*, pp. 49–59, Springer, 2016.

[4] A. Ultsch, "Clustering wih som: U* c," in *Proceedings of the 5th Workshop on Self-Organizing Maps*, vol. 2, pp. 75–82, 2005.

[5] G. Pölzlbauer, "Survey and comparison of quality measures for self-organizing maps," in *Proceedings of the Fifth Workshop on Data Analysis (WDA-04)*, pp. 67–82, Elfa Academic Press, 2004.

[6] R. Mayer, R. Neumayer, D. Baum, and A. Rauber, "Analytic comparison of self-organising maps," in *Advances in Self-Organizing Maps*, pp. 182–190, Springer, 2009.

[7] A. Ultsch, "Self organized feature maps for monitoring and knowledge aquisition of a chemical process," in *ICANN'93*, pp. 864–867, Springer, 1993.

[8] L. Hamel and C. W. Brown, "Improved interpretability of the unified distance matrix with connected components," in *7th International Conference on Data Mining (DMIN'11)*, pp. 338–343, 2011.

[9] H. Yin and N. M. Allinson, "On the distribution and convergence of feature space in self-organizing maps," *Neural computation*, vol. 7, no. 6, pp. 1178–1187, 1995.

[10] I. Miller and M. Miller, *John E. Freund's Mathematical Statistics with Applications (7th Edition)*. Prentice Hall, 7 ed., 2003.

[11] L. Hamel and C. W. Brown, "Bayesian probability approach to feature significance for infrared spectra of bacteria," *Applied Spectroscopy*, vol. 66, no. 1, pp. 48–59, 2012.

[12] L. Hamel and B. Ott, "A population based convergence criterion for self-organizing maps," in *Proceedings of the 2012 International Conference on Data Mining*, (Las Vegas, Nevada), July 2012.

[13] E. Merényi, K. Tasdemir, and L. Zhang, "Learning highly structured manifolds: harnessing the power of SOMs," in *Similarity-based clustering*, pp. 138–168, Springer, 2009.

[14] T. Villmann, R. Der, M. Herrmann, and T. M. Martinetz, "Topology preservation in self-organizing feature maps: exact definition and measurement," *Neural Networks, IEEE Transactions on*, vol. 8, no. 2, pp. 256–266, 1997.

[15] K. Kiviluoto, "Topology preservation in self-organizing maps," in *IEEE International Conference on Neural Networks*, pp. 294–299, IEEE, 1996.

[16] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[17] D. Beaton, I. Valova, and D. MacLean, "Cqoco: A measure for comparative quality of coverage and organization for self-organizing maps," *Neurocomputing*, vol. 73, no. 10, pp. 2147–2159, 2010.

[18] R. Wilcox, "Kolmogorov–smirnov test," *Encyclopedia of biostatistics*, 2005.