Improving the Performance of Self-Organizing Maps via Growing Representations

Mathew Merkow and Robert Kirk DeLisle*

Computational Research, Array BioPharma, Inc, 3200 Walnut Street, Boulder, Colorado 80501

Received April 23, 2007

Self-organizing maps (SOMs) are a type of artificial neural network that through training can produce simplified representations of large, high dimensional data sets. These representations are typically used for visualization, classification, and clustering and have been successfully applied to a variety of problems in the pharmaceutical and bioinformatics domains. SOMs in these domains have generally been restricted to static sets of nodes connected in either a grid or hexagonal connectivity and planar or toroidal topologies. We investigate the impact of connectivity and topology on SOM performance, and experiments were performed on fixed and growing SOMs. Three synthetic and two relevant data sets from the chemistry domain were used for evaluation, and performance was assessed on the basis of topological and quantization errors after equivalent training periods. Although we found that all SOMs were roughly comparable at quantizing a data space, there was wide variation in the ability to capture its underlying structure, and growing SOMs consistently outperformed their static counterparts in regards to topological errors. Additionally, one growing SOM, the Neural Gas, was found to be far more capable of capturing details of a target data space, finding lower dimensional relationships hidden within higher dimensional representations.

INTRODUCTION

Cheminformatics techniques involving large numbers of chemical descriptors have become widely applied within chemistry-based fields of research and discovery. While early descriptions of chemical structures such as molecular weight, log P,[1] and substructural content are still used extensively, numerous additional representations have been proposed. The DRAGON descriptor calculation software, for example, currently computes in excess of 1600 individual descriptors, and more are added regularly.[2,3] Developments in the fields of machine learning and artificial intelligence over the past two decades have provided tools applicable to the complexities of chemical space beyond simple linear regression and linear discriminant anlyasis.[4,5] Additionally, tools are now available that can handle the potentially high dimensional space of the chemistry domain as well as provide regression and classification beyond linear methodologies.

Self-organizing maps (SOMs)[6,7] have become more prevalent in a variety of modeling tasks over the past few years. One of the first applications of SOMs was to map molecular surface properties from three-dimensional space to two-dimensional maps.[8] Typically, these consisted of projections of molecular surface points to the surface of a sphere while maintaining the pairwise distances between points as closely as possible to their original 3-D distances. These maps could then be used more readily for visualization and analysis of surface property changes with respect to activities. A brief survey of the recent chemistry literature within the pharmaceutical domain reveals application of SOMs to a variety of problems, including clustering of the NCI anticancer database of compounds on the basis of the GI50 values across the NCI-60 cell lines to reveal mechanisms of action,[9,10] virtual screening and selection of diverse and representative subsets of screening compounds,[11,12] and evaluation of compound libraries for desired activites[13] and selectivity.[14] Applications within the bioinformatics literature have recently included alignment-free G-protein-coupled receptor classification,[15] and improving sequence alignments by inclusion of information gleaned from SOMs,[16] among others. While SOMs were originally designed (and are still typically applied) as unsupervised methods, supervised SOMs have been developed and applied to traditional QSAR problems[17] and extended with simulated annealing to provide feature selection.[18]

Unlike standard clustering algorithms such as K-means or hierarchical agglomerative clustering, the cluster representatives within an SOM are closely interrelated and have an impact on one another. During the training process, entire regions or neighborhoods within the map are affected, thus providing not only a clustering of the data but an evaluation of the relationships between clusters in the original data space. The typical implementation of an SOM defines an underlying connectivity and topology that are used during the training process to define how neighborhoods are evaluated. Square and hexagonal connectivities are common and result in each node having four or six neighbors within the SOM, respectively. The topological arrangement of the nodes within the SOM are often planar, torroidal, or, as mentioned above, spherical. While the planar topology is common, toroidal topologies are often used in which the extreme edges of the planar map are connected, providing a continuous surface which is by definition free of edge effects. Creating a spherical topology also removes edge effects, but it is more complex due to the need to define neighborhoods that are appropriately connected. (Spherical topologies are not considered here.) It is not clear, however, whether either a toroidal or spherical surface is appropriate for the representation of chemical space, as this space does not "wrap" in any meaningful sense.

* Corresponding author. E-mail: rkdelisle@earthlink.net.

10.1021/ci7001445 CCC: $37.00
Published on Web 08/18/2007

SOMs' ability to provide information beyond the simple identification of clusters within data and extending to the actual relationships between clusters can be further utilized. An analysis of the distances between nodes or clusters of an SOM trained on corporate databases of compounds, for example, can be used to identify holes within the collection that may be filled by commercially available or collaborator compounds, or internal synthetic efforts. Further, the trained SOM can simplify the comparison of two independent collections of compounds significantly. Rather than requiring a comparison of all possible pairs in each collection, the most similar node to a query compound can be identified easily. Subsequently comparing all those structures associated with the nearest SOM node and its immediate neighbors to the query compound will identify the closest matching compound from the modeled database. This type of searching can reduce the number of necessary comparisons by orders of magnitude. Also, compounds within the corporate collection that are most similar to a query compound of interest can be identified rapidly. Unlike traditional clustering methods, the neighborhood property of SOMs can be used to evaluate those compounds belonging not only to the closest cluster but to adjacent and typically also similar clusters. All of these techniques are dependent, however, upon an accurate representation of cluster relationships within the underlying data. If a distortion of the underlying topology is present within the trained SOM, clusters that are neighbors within the SOM topology may be less related than more distant clusters, leading to interpretational errors.

In order to investigate the impact of the chosen topology and connectivity on the performance of SOM modeling, we performed a series of experiments using different topologies. In addition to topologies of a predefined dimension, we introduce the application of growing topologies, which has not yet been seen in the cheminformatics literature. This method is unique in that a minimal map is used at the beginning of training and is effectively grown into the data space as needed. The result is a map that conforms to the modeled space rather than assuming the chosen topology and connectivity are correct, and we show that growing topologies may be more appropriate than defined topologies for some problems in cheminformatics. Furthermore, an algorithm with the ability to adapt the degree of connectivity of nodes within the SOM (the Neural Gas algorithm) is shown to minimize topological and quantization errors in comparison to all others tested.

METHODS

Data Sets. Three synthetic data sets and two real data sets were used for an evaluation of SOM methods. The synthetic data sets were constructed to represent idealized data distributions in which data points form (a) well isolated clusters, (b) differing densities of data points in different regions of the data space, or (c) difficult to represent structures within the data space. Two real data sets were also chosen that have very different dimensionalities and are drawn from different chemistry application domains.

The Rectangles and Star data sets (Figure 1) are composed of 1044 and 4990 two-dimensional data points, respectively, with both dimensions falling in the domain (0,1). The Rectangles data were generated to produce four disjoint rectangles each with a different density of data points. The Star data were generated to produce a star shape having different densities of data in the star's five points and central body. Data points were randomly drawn from uniform probability distributions constrained by the edges of the rectangles or star.

The MultiD data set consists of 810 three-dimensional points with all dimensions in the domain (0,1). Within the three-dimensional space of the data set, three distinct subdimensionalities are represented: a three-dimensional, rectangular block connected to a two-dimensional plane by a linear set of points (Figure 1). This data set thus provides independent one-, two-, and three-dimensional components all embedded with a three-dimensional space, and a significant modeling challenge.

The real data sets enable us to test the effects of varying SOM topologies and connectivities with data relevant to typical modeling activities. The Blood-Brain Barrier (BBB) data set was taken from the literature,[19] and the Estrogen Receptor (ER) data set was kindly provided by Dr. Weida Tong of the National Center for Toxicological Research.[20] The BBB data set is composed of nine computed descriptors for 325 chemical compounds, and the ER data set consists of 232 chemical compounds and 197 computed descriptors. Descriptors were normalized (adjusted to have a mean of 0 and standard deviation of 1) in order to prevent effects induced by differing scales. The measured CNS-penetration or ER-binding capability are also supplied for each compound as binary endpoints, but these values were not used for these experiments. While SOMs can be extended beyond their typical application as unsupervised modeling techniques, this was beyond the scope of these experiments.

Self-Organizing Maps. The basic algorithm used for self-organizing maps is based upon that described by Fritzke.[21,22] The connectivity, topology, and extents of the map are decided upon, and then each node of the map is initialized by either setting the weights to random values clamped by the minimum and maximum of each dimension or by using the data values of randomly selected observations from the data set. Each node within the map then has a dual representation: (1) a node's weights define its location within the data space, and these weights are used for comparison to observations from the training data; (2) each node has a location within the topology of the SOM and has neighbors defined by the connectivity. Iterative training of the map proceeds as follows.

Parameters:

t = time parameter or iteration counter, initialized to 0

t_{max} = maximum number of iterations to perform

ϵ_i = initial node adaptation

ϵ_f = final node adaptation at t_{max}

σ_i = initial standard deviation of the Gaussian defining the neighborhood

σ_f = final standard deviation of the neighborhood Gaussian at t_{max}

1. An observation, ξ, is selected at random from the data set.

2. The winning node in the map (s) is that node closest to the selected observation. In these experiments, the Euclidean distance was used as the distance metric.

3. The weights, $\mathbf{w}$, of each node, r, in the map are adapted by

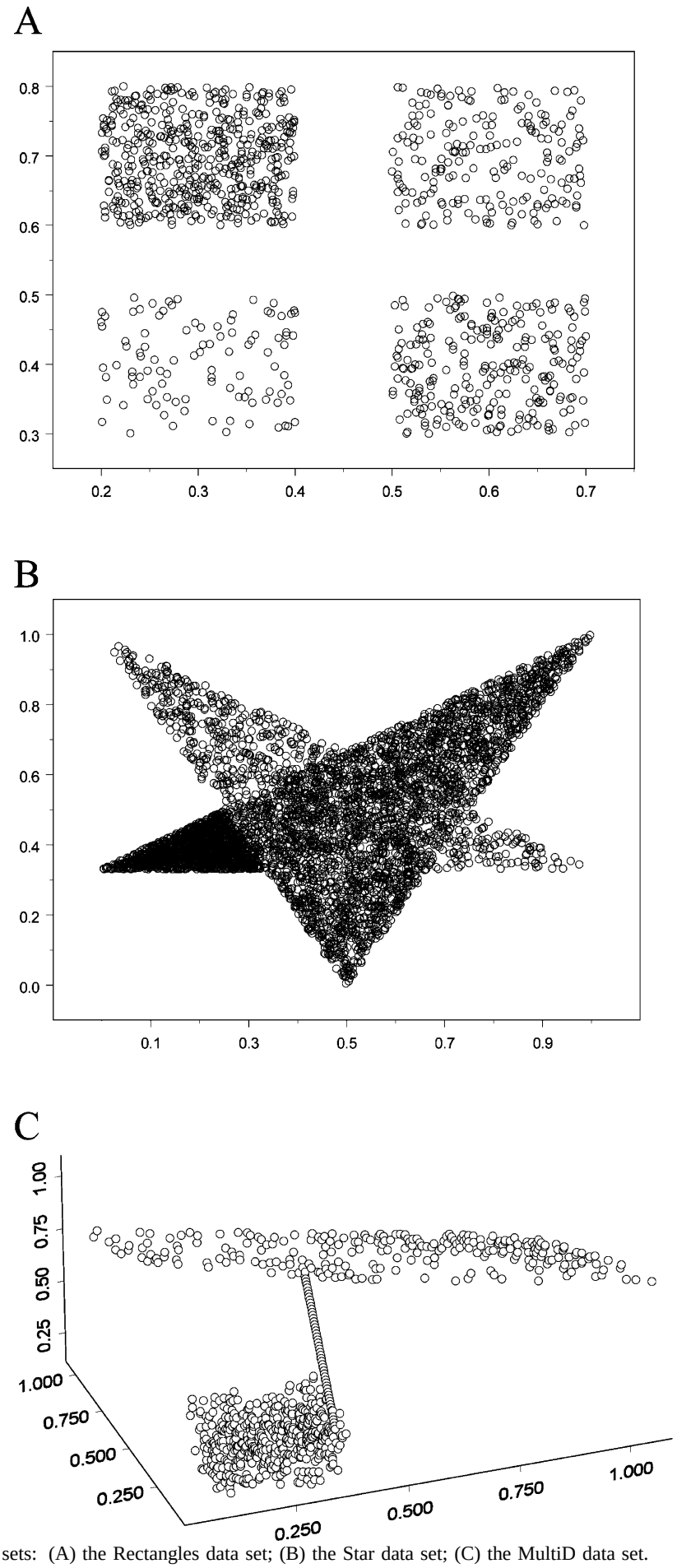

Figure 1. Synthetic data sets: (A) the Rectangles data set; (B) the Star data set; (C) the MultiD data set.

$$\Delta \mathbf{w}_r = \epsilon(t) h_{rs} (\mathbf{w}_\xi - \mathbf{w}_r) \quad (1)$$

with

$$\epsilon(t) = \epsilon_i \left(\frac{\epsilon_f}{\epsilon_i}\right)^{t/t_{max}} \quad (2)$$

$$h_{rs} = \exp\left[\frac{-d(r,s)^2}{2\sigma(t)^2}\right] \quad (3)$$

$$\sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i}\right)^{t/t_{max}} \quad (4)$$

$$d(r,s) = \text{distance between nodes } r \text{ and } s \text{ within the SOM} \quad (5)$$

4. Increment the time parameter: $t = t + 1$.

5. If $t < t_{max}$, continue from step 2.

The parameters ϵ and σ control the degree of adaptation applied to node weights. $\epsilon(t)$ is a time-dependent (or iteration-dependent) exponential decay function that decreases from ϵ_i to ϵ_f over t steps. Likewise, $\sigma(t)$ is an exponential decay function used to control the width of the Gaussian h_{rs} defining the neighborhood adjustment window. The effect is that, during each adaptation step, the degree of change to any particular node's weights decreases with the distance from the winning node, *s*. The dropoff due to distance from *s* is more pronounced over time as a result of the decaying factor σ, and the degree of change is further decreased in a time-dependent fashion by the factor ϵ.

Rectangular Grid. The most basic topology for a self-organizing map is a two-dimensional, rectangular grid in which a planar topology is used. Every node in the map (apart from those at the edge of the map) has four neighbors to which it is directly connected. The distance between nodes of eq 5 is defined here as the Manhattan distance:

$$d(r_{ij}, s_{km}) = |i - k| + |j - m| \quad (6)$$

where (*i*,*j*) *and* (*k*,*m*) are the coordinates of *r* and *s*, respectively, within the two-dimensional grid. The Manhattan distance effectively provides the connection distance (number of edges between nodes in the topology of the SOM) and is consistent with the distance measure used for the hexagonal connectivity, described below. Any valid distance can be used, however.

Torus. A common topology used in SOMs connects opposite edges of a grid to form a toroidal map, effectively eliminating edges and creating a continuous two-dimensional surface. The Manhattan distance can still be used with appropriate consideration of the wrapping effect to ensure the shortest distance between nodes is found.

Hexagonal Grid. Similar to the Rectangular Grid; however, each node in the Hexagonal Grid (apart from those at the edge of the map) has six neighbors to which it is directly connected. The distance metric for the hexagonal grid is defined as the smallest number of edges that can be traversed while traveling between two nodes.

Growing Hex. The Growing Hexagonal SOM is similar to the Hexagonal SOM in that they have the same topology, connectivity, and distance metric. The training algorithm requires some extra bookkeeping to incorporate growth. After a winning node (*s*) is selected, the accumulated error, δ, is increased by the Euclidian distance (2-Norm) between *s* and the current observation. After a specified number of training iterations the **Hexagonal Grow** operation is called.

Hexagonal Grow Algorithm

1. Find the node, *g*, in the map with the largest accumulated error.

$$g = \text{argmax}_{\forall n \in A}[\delta(n)] \quad (7)$$

2. If *g* does not have a complete set of neighbors, add new nodes around *g* to complete its set of neighbors and fill in edges accordingly.

3. Evenly distribute the error in *g* to its neighbors.

4. Reset the accumulated error of *g* to zero.

Growth also affects the completion criterion for the SOM. In order to compare the different types of SOM, the ratio between the number of observations in the data set and the number of nodes in the SOM needs to be approximately the same. A growing SOM is complete when this ratio reaches a specified threshold, for example, one node per 10 observations in the modeled data set.

Neural Gas. The Neural Gas is a very different type of SOM. It has no defined topological structure or connectivity and can grow/remove nodes/edges during training. The Neural Gas SOM requires some extra notation:

·Let us define a node as a basic element of the self-organizing map.

·Each node, *n*, has a weight, w_n, which represents its position in the data space of the training data.

·Each node will also have an accumulated error $\delta(n)$, which is a real number between zero and infinity.

·Let us define an edge $\gamma_{n_1-n_2} \equiv \gamma_{n_2-n_1}$ as the bidirectional connection between nodes n_1 and n_2 in the map.

·Each edge $\gamma_{n_i-n_j}$ has an age such that $age(\gamma_{n_i-n_j})$ is a integer between zero and infinity, which will be written as $age(n_i,n_j)$ for expediency.

·Let N_r be the set of nodes directly connected to *r* or the neighborhood of node *r* defined as

$$N_r = \{n_1, ..., n_i\} \text{ such that } \exists\ \gamma_{r-n_k} \text{ for each } k = 1, ..., i \quad (8)$$

Neural Gas Training Algorithm

Parameters:

t = time parameter or iteration counter, initialized to 0

t_{grow} = number of training iterations before growth occurs

a_{max} = maximum age allowed for an edge

β = error reduction value

ϵ_i = initial node adaptation

ϵ_f = final node adaptation at t_{max}

σ_i = initial standard deviation of the Gaussian defining the neighborhood

σ_f = final standard deviation of the neighborhood Gaussian at t_{max}

ρ = current data set to node ratio of the growing map

ρ_f = target data set to node ratio, completion criterion

1. Initialize SOM by creating a set of nodes, *A*, that contains two nodes n_1 and n_2

$$A = \{n_1, n_2\} \quad (9)$$

Weights are assigned for each node by choosing random observations from the data set and assigning the values of their descriptors to the weights of w_{n1} and w_{n2}. Create a set of edges, ψ, that contains an edge connecting n_1 and n_2.

$$\psi = \{\gamma_{n_1-n_2}\} \tag{10}$$

2. An observation, ξ, is selected at random from the data set.

3. Determine the two closest nodes to ξ, s_1 and s_2, such that

$$s_1, s_2 \in A \tag{11}$$

$$s_1 = \arg\min_{\forall n \in A} |\xi - n| \tag{12}$$

$$s_2 = \arg\min_{\forall n \in A\{s_1\}} |\xi - n| \tag{13}$$

$$|\xi - n| = \text{Euclidian distance between } \xi \text{ and } n \tag{14}$$

4. If there is an edge $\gamma_{s_1-s_2}$, set $age(s_1,s_2)$ to zero; if $\gamma_{s_1-s_2}$ does not exist, create $\gamma_{s_1-s_2}$.

5. Update error $\delta(s_1)$, such that

$$\delta(s_1) = \delta(s_1) + |\xi - n|^2 \tag{15}$$

6. Adjust the weights, w, of s_1 and each node $r \in N_{s1}$ using eq 1, with

$$\epsilon(t) = \epsilon_i\left(\frac{\epsilon_f}{\epsilon_i}\right)^{\rho/\rho_f} \tag{16}$$

$$\sigma(t) = \sigma_i\left(\frac{\sigma_f}{\sigma_i}\right)^{\rho/\rho_f} \tag{17}$$

7. Increment the age of all edges connected to s_1:

$$age(s_1,i) = age(s_1,i) + 1(\forall i \in N_{s_1}) \tag{18}$$

8. Remove edges with $age()$ older than a_{max}. If this process results in nodes that are no longer connected to any other nodes, remove these nodes as well.

9. If the number of training iterations is greater than t_{grow}, start the **Neural Gas Grow** algorithm and reset t to zero.

10. Reduce the error of all nodes by

$$\delta(n_i) = \delta(n_i)(1 - \beta), (\forall n_i \in A) \tag{19}$$

11. If ρ has not been reached, continue from step 2.

Neural Gas Grow

Parameter:

α = error reduction coefficient

1. Find the node, r, with the largest accumulated error:

$$r = \arg\max_{\forall n \in A}[\delta(n)] \tag{20}$$

2. Find the node, $s \in N_r$, with the largest error:

$$s = \arg\max_{\forall n \in N_r}[\delta(n)] \tag{21}$$

3. Add a new node, z, with weights given by

$$w_z = \frac{(w_r + w_s)}{2} \tag{22}$$

4. Add edges γ_{r-z} and γ_{s-z}, remove γ_{r-s}.

5. Decrease the error of r and s:

$$\delta(n_r) = \delta(n_r)(1 - \alpha) \tag{23}$$

6. Set the error of new node z:

$$\delta(n_s) = \delta(n_s)(1 - \alpha) \tag{24}$$

$$\delta(n_z) = \frac{\delta(n_r) + \delta(n_s)}{2} \tag{25}$$

Error Measures. Two error measures were selected to evaluate the overall performance of each SOM generated. Quantization error (QE, eq 26) measures the degree to which the SOM represents, or covers, data points within the underlying data space.

$$\text{QE} = \frac{\sum_{i=1}^{N} D(w_i,w_s)}{N} \tag{26}$$

with N = number of data points in the data set and $D(w_i,w_s)$ = Euclidean distance between the data point i and the weights of the closest node, s, in the SOM.

Quantization error is directly dependent upon the number of dimensions of the data set and the scale of the descriptors. The actual value may not necessarily be comparable between data sets due to the Euclidean distance tending to result in higher values for $D(w_i,w_s)$ in higher dimensional spaces, and the impact of descriptor scales.

Topological error (TE, eq 27) measures the degree of topological distortion resulting from mapping the data set with the SOM. TE is unique in that it does not consider the quantitative representation of the underlying data space as does QE but rather addresses the degree to which neighboring nodes in the SOM are representative of neighboring points or regions within the original data space.

$$\text{TE} = \frac{\sum_{i=1}^{N} \begin{matrix}1, \text{ if } s_1 \text{ and } s_2 \text{ are not connected} \\ 0, \text{ if } s_1 \text{ and } s_2 \text{ are connected}\end{matrix}}{N} \tag{27}$$

with s_1 being the closest node to the sampled data point and s_2 being the second closest. Observant readers will notice that TE can be effectively reduced to zero simply by making the SOM fully connected, that is, every node sharing an edge with every other node within the SOM. For this reason, we also include the average number of connections per node as (1) validation that the growing process has not simply added an excessive number of connections, thus eliminating the usefulness of the TE, and (2) a measure of the resulting complexity of the trained SOM. Note that, for the static SOMs, this value will be equivalent for maps of identical size and connectivity, with square connectivities being slightly less than four, and hexagonal connectivities being slightly less than six. (Torroidal topologies will have values of four or six for rectangular or hexagonal connectivites due

to the elimination of corners and edges within the map.) Using these TE values as comparison standards will provide an interesting evaluation of the growing methodologies' ability to accurately represent the modeled data space.

Parameter Optimization. It is clear from the description of the SOM implementation that the overall performance of the technique is impacted by numerous adjustable parameters. While most descriptions of SOM usage supply details of the specific parameters used, very few supply details of the optimization of those parameters. In order to ensure optimal performance of each topological method and thus allow valid comparisons between them, parameters were optimized for each topology independently. Following is a description of the methodology used to optimize SOM parameters, and additional details can be found in the Results and Discussion.

The Rectangular Grid, Toroid, Hexagonal Grid, and Growing Hex are the simplest SOMs with the fewest adjustable parameters. Still, each method requires setting their dimensions, $t_{max,}$ ϵ_i, ϵ_f, σ_i, and σ_f, appropriately. Of these, ϵ_f and σ_f represent the adaptation and neighborhood windows at the end of the training process and should thus simply be set arbitrarily small. A value of 0.01 was chosen for ϵ_f, representing a maximum of 1% change in node weights during the adaptation step, and this is further reduced by multiplication with the neighborhood window factor during the adaptation step. For σ_f, a value less than 1.0 will lead to no neighborhood effect at the end of training. A value of 0.5 was chosen for σ_f in order to ensure that, during the final steps of training, a negligible neighborhood effect was present. This would allow centering of each node within its local region of space with no external disturbances emanating from adjacent nodes.

The number of nodes in each map was arbitrarily set to be 10% of the number of observations in the training set: $\rho = 0.1$. This was done in order to provide obvious data reduction and allow evaluation of the impact of this reduction on interpretation of the original data space. For static SOMs, the dimensions were adjusted to provide as close as possible the target number of nodes. (For example, the Rectangles data set consists of 1044 observations, leading to 104 nodes in the growing maps. The static SOMs were thus set to 10×10 resulting in 100 nodes.)

Setting t_{max} must take into consideration the degree of overall adaptation desired for the map and often can be considered as a multiple of the total number of observations in the training data. The growing topologies are not affected by t_{max} but rather define their terminations by acquiring a specified number of nodes. This can lead to extended training times that are not predeterminable. To address this, an upper estimate of the number of iterations required by the growing topologies was determined, and the static topologies were then optimized using this upper limit as t_{max}, ensuring a similar degree of training for each.

For the static SOMs and the Growing Hex, the initial adaptation and neighborhood values ϵ_i and σ_i were optimized using a grid search. Multiple values of σ_i were chosen to represent very narrow to maximally wide neighborhoods. For example, the Rectangles data set required ~100 nodes and thus a 10×10 map—chosen values for σ_i were 1, 5, and 10. At each of these values for σ_i, ϵ_i was optimized by training 10 independent SOMs with ϵ_i values ranging from 0.05 to 0.95 in steps of 0.05. An optimal value for ϵ_i was chosen on the basis of these results, and subsequently, σ_i was optimized in a similar fashion, that is, adjusting σ_i from 0.5 to the maximal value in steps of 0.5, and training 10 independent SOMs at each step.

The Neural Gas has three additional parameters requiring attention: α, β, and AGE. It was found empirically that β merely needed to be set arbitrarily small, and a value of 0.0005 was selected. Rather than attempting a four-dimensional grid search to optimize σ_i, ϵ_i, α, and a_{max}, the optimization results from the static topologies were exploited to provide values for σ_i and ϵ_i, allowing the additional parameters to be addressed.

Implementation. The SOM application was developed in-house and written in C++. The cross-platform, open-source IDE Code::Blocks (www.codeblocks.org) was used for code development, and GCC 3.4.5 (gcc.gnu.org) was used as the compiler on either Windows or LINUX. (The code was developed to run on both platforms.) Application timings were not performed. In general, growing maps took somewhat longer than static maps, but even the largest data sets finished within a maximum of 2−3 h.

RESULTS AND DISCUSSION

For all analyses, the chemical data and synthetic data performed similarly. Changes observed for QE and TE during optimization of all parameters were consistent across data sets, and the relative performances of the various topologies analyzed were also consistent. Given that the synthetic data sets are two- and three-dimensional, whereas the chemical data sets are 9- and 197-dimensional, it was not initially clear that findings on the synthetic data sets would translate to real-world data. This uncertainty was initially supported by the uniformity of construction of the synthetic data sets. Given the results discussed below, we see no evidence that either type of data could be modeled any better or worse by the SOMs. This further suggests that the results described here should be transferable to other data sets and domains.

All three static SOMs and the Growing Hex SOM behaved similarly during the optimization of ϵ_i (Figure 2). Generally, both error measures were reduced and stabilized by $\epsilon_i = 0.20-0.30$, with further increases having little or no effect on either QE or TE. This comes as no surprise as ϵ_i partially controls the degree of adaptation of SOM nodes and small values would lead to very little adaptation to the modeled data space. Larger values for ϵ_i are tempered by the decay function which reduces the actual level of adaptation during later stages of SOM training. TE represents the fraction of observations in the data set for which the two nearest SOM nodes are not connected, and the impact on TE is much more striking. Across data sets, this value was consistently seen to drop significantly as ϵ_i increased; for example, for the Star data set, initial values of 40% topological error are reduced to ~25% for the Rectangular Grid topology (Figure 2). A value of 0.5 was chosen for ϵ_i as it seemed to ensure stabilization of errors and exhibited no detrimental effects across all data sets tested.

The optimization of σ_i followed a similar trend as that of ϵ_i, with decreases in both errors as σ_i increased (Figure 3). QE exhibited a modest rebound in error at higher values of σ_i. Larger values of σ_i extend the neighborhood effect during the SOM algorithm's adaptation step, thus delocalizing the

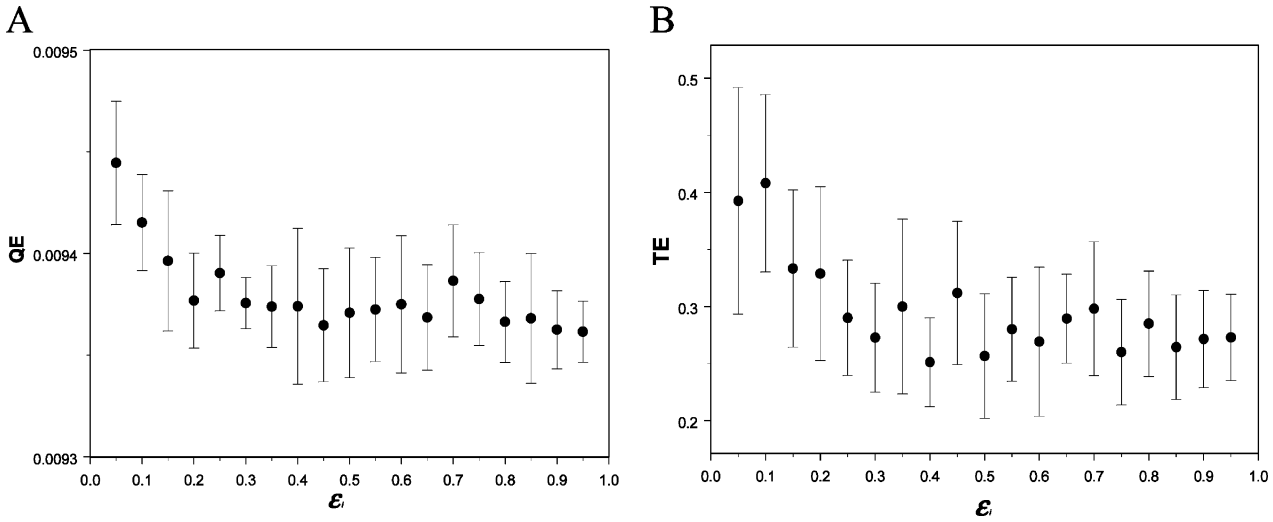

Figure 2. Effect of adjusting ϵ_i on QE and TE with σ_i held constant for the Star data set. The Rectangular Grid was used, but similar results were seen with all static and the Growing Hex. Shown are the mean and standard deviation of error measures for 10 independently trained SOMs at each point.

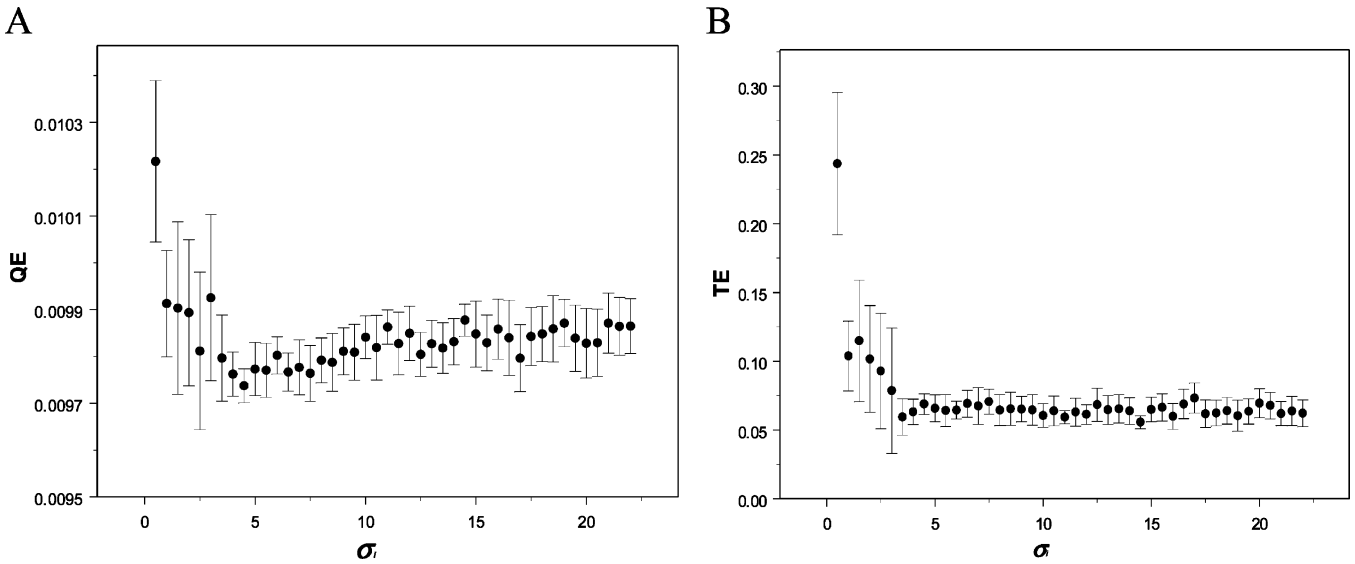

Figure 3. Effect of adjusting σ_i on QE and TE with $\epsilon_i = 0.5$ for the Star data set. The Hexagonal Grid was used, but similar results were seen with all the static SOMs. Shown are the mean and standard deviation of error measures for 10 independently trained SOMs at each point.

adaptation and effectively pulling the neighborhood of nodes toward a particular centroid, thus shifting it from its own data space. The effect on TE was again more striking, with large reductions as σ_i increased, but with no rebound at higher σ_i values. From these results, it appears that setting σ_i to ~25% of the expected width of the map minimized TE without leading to excessive rebound in QE. This seemed consistent across maps varying in size from 4×4 to 22×22 nodes; however, it may likely be data-set- and topology-dependent, requiring individual optimization.

The Neural Gas SOM required optimization of two additional parameters: α and a_{max}. The parameter α controls the degree of error reduction after a growth step and affects all nodes connected to the new node. Theoretically, this localized error reduction helps to dampen growth in the area of the map in which nodes were most recently added. This allows all regions opportunities to grow into their data spaces as necessary. The impact on QE and TE appeared to be either nonexistent or minimal with a negligible reduction in error that stabilized quickly (data not shown). It would appear that the selection of a value for α could be made somewhat arbitrarily, and a value of 0.5 was settled upon.

a_{max} controls the rate at which links between nodes are removed from the growing topology. During each iteration, the age of all links connected to the closest node are incremented, while the age of the link between the closest and second closest nodes in the map to the current data point is reset to 0. This reinforces the importance of this link and prevents its removal. If the age of a particular link exceeds a_{max}, it is removed, thus reducing any unnecessary neighborhood effects and allowing the previously connected nodes to adapt independently. A very low value for a_{max} leads to a map in which very few links exist, and the nodes fall into numerous, disjoint microclusters, whereas a high value will lead to an overconnected map with excessive neighborhood effects. Optimizing a_{max} led to differential responses of QE and TE, with TE typically being minimized between 40 and 80, while QE increased slightly across all values (Figure 4). As seen before, the increase in QE is a result of an enhanced neighborhood effect leading to a centering of the map and

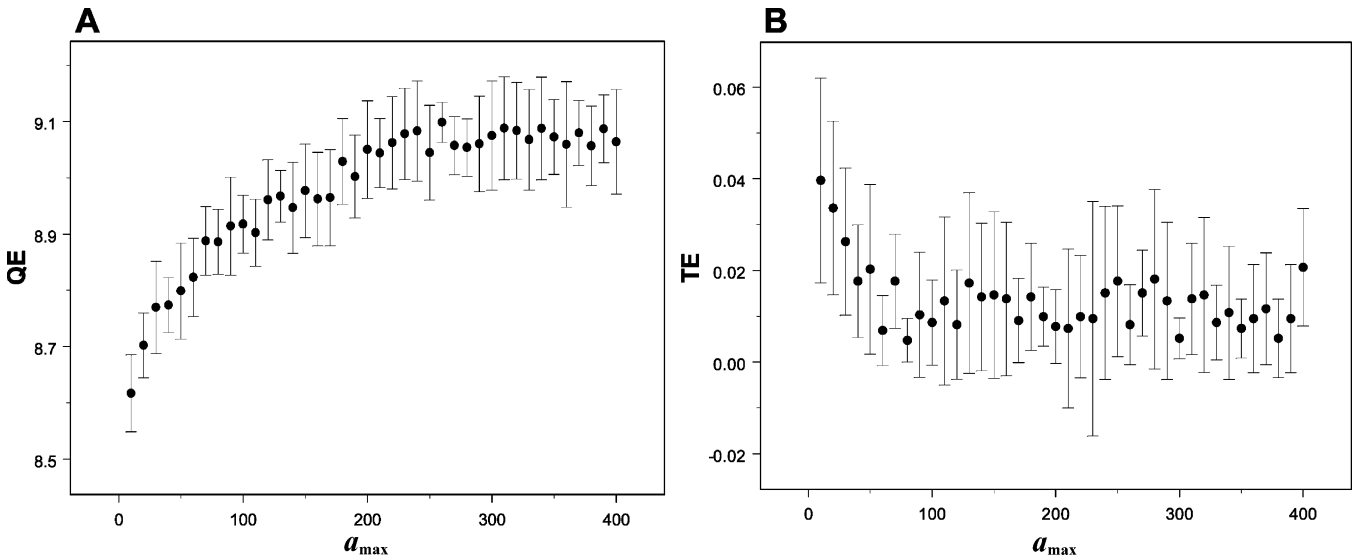

Figure 4. Effect of adjusting a_{max} on QE and TE for the ER data set using the Neural Gas topology. Shown are the mean and standard deviation of error measures for 10 independently trained SOMs at each point.

Table 1. Quantization Error Values[a]

	Rectangles	Star	MultiD	BBB	ER
Rectangular Grid	0.015	**0.009**	0.569	**1.044**	8.676
	(<0.001)	**(<0.001)**	(0.005)	**(0.009)**	(0.063)
Toroid	0.017	0.010	0.625	1.099	8.868
	(<0.001)	(<0.001)	(0.010)	(0.007)	(0.033)
Hexagonal	0.016	0.010	0.607	1.086	8.813
	(<0.001)	(<0.001)	(0.014)	(0.011)	(0.059)
Growing Hex	0.016	**0.009**	0.541	1.086	8.822
	(<0.001)	**(<0.001)**	(0.010)	(0.009)	(0.055)
Growing Neural	**0.014**	**0.009**	**0.499**	1.047	**8.630**
Gas	**(<0.001)**	**(<0.001)**	**(0.006)**	(0.012)	**(0.063)**

[a] Quantiazation error was computed by eq 26. Values presented are the lowest, average (standard deviation) values seen across optimization runs, with 10 SOMs run at each step (see *Parameter Optimization*). The lowest values for each data set are in bold.

Table 2. Topological Error Values[a]

	Rectangles	Star	MultiD	BBB	ER
Rectangular Grid	0.040	0.261	0.126	0.150	0.101
	(0.005)	(0.046)	(0.011)	(0.035)	(0.032)
Toroid	0.457	0.562	0.182	0.321	0.208
	(0.045)	(0.071)	(0.025)	(0.028)	(0.033)
Hexagonal	0.020	0.056	0.073	0.086	0.083
	(0.007)	(0.005)	(<0.001)	(0.015)	(0.044)
Growing Hex	**0.001**	0.025	0.037	0.014	0.015
	(0.001)	(0.008)	(<0.014)	(0.016)	(0.007)
Growing Neural	0.003	**0.002**	**0.002**	**0.009**	**0.005**
Gas	(0.003)	**(<0.001)**	**(0.003)**	**(0.004)**	**(0.005)**

[a] Topological error was computed by eq 27. Values presented are the lowest, average (standard deviation) values seen across optimization runs, with 10 SOMs run at each step (see *Parameter Optimization*). The lowest values for each data set are in bold.

increase in error at the SOM's periphery. The a_{max} parameter showed some consistency across data sets, but its optimal value may be different between data sets and will likely require individual optimization.

Comparing values for QE between SOM topologies shows that all those investigated here are capable of quantizing the underlying data space equally well (Table 1). While some variation occurs across topologies, the optimized values are quite similar and certainly not significantly different enough to suggest one topology better represents the data than any other.

Differences in TE between SOMs, and in particular between static and growing topologies, are, however, quite dramatic (Table 2). In general, the static maps require proper optimization of σ_i in order to prevent twisting of the map within the data space (Figure 5). This twisting results from the SOM not properly adapting to the modeled data space and locating nodes that are not topologically connected within the SOM near each other in the modeled space. This is most often seen with a value for σ_i that is too small and thus not enforcing proper neighborhood adaptation to allow the map to unfold. The effect of twisting is also seen as a significant increase in TE. In the case of Figure 5, when hexagonally connected SOMs are trained with $\sigma_i = 1$, the resulting TE values are $\sim 0.17 \pm 0.06$ (average and standard deviation of 10 trained maps), compared to the value of 0.02 ± 0.007 when $\sigma_i = 3$ or higher (see Table 2). The growing SOMs are not subject to twisting as they are initialized with a minimum number of nodes and grown into the data space as needed. The nature of the adaptation step does not lead to any severe topological distortions of this type, and thus the growing SOMs avoid this pitfall.

Among the static maps, the Hexagonal Grid (hexagonal connectivity and planar topology) generally produced the lowest TE values, while values for the Torus (square connectivity and toroidal topology) were alarmingly high. The ability of the hexagonal representation to produce a low TE can be explained by the fact that there exists a triangular subconnectivity within each hexagon, as well as for edge nodes that are not fully connected. The result is that, for any data observation occurring inside a triangular region, the closest and second closest nodes will by definition be connected to each other (Figure 6). In fact, if the SOM was completely free of twisting and it completely covered the data, TE would be zero. We do not see this occur due to the fact that the hexagonal SOMs are contained within the convex hull of the data, and thus there exist points in the data that are outside the boundaries covered by the SOM. The closest and second closest nodes to an observation from the data set are not necessarily connected within the

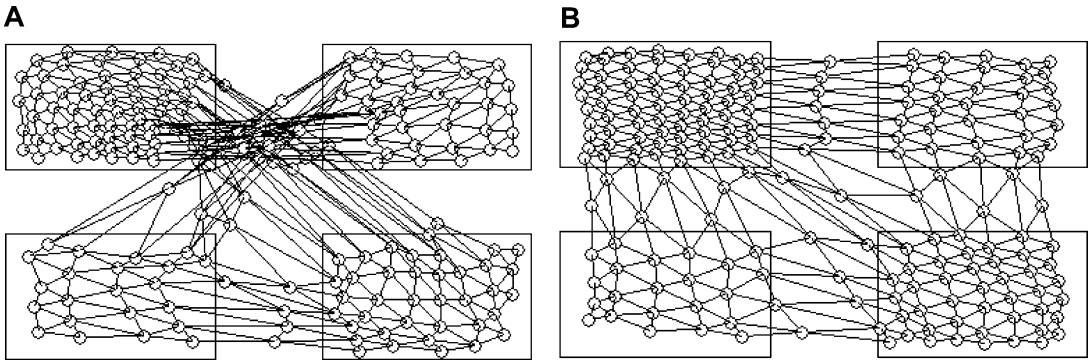

Figure 5. Twisting within the SOM topology. The Hexagonal Grid was trained on the Rectangles data set setting σ_i to 1 and yielding TE = 0.17 (A) or to 3 and yielding TE = 0.02 (B). The rectangles illustrate the locations of data points with the upper left having low density, lower left having high density, and the two right rectangles each having a medium density of points. Similar results were seen for the Star and MultiD data sets.

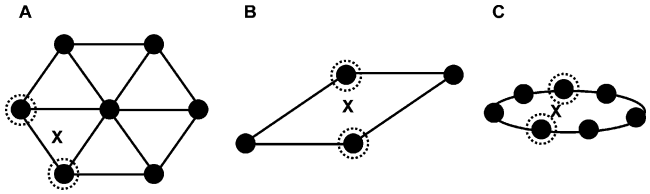

Figure 6. Contributors to TE in differing SOMs. A sampled data point is depicted by **X**, and the two closest nodes highlighted. (A) The basic hexagonal representation is composed of triangular subconnectivities. This allows the closest and second closest nodes in the map to be connected. (B) The basic rectangular connectivity is prone to warping, which causes nodes closest to the sampled data point not to be directly connected. (C) Toroidal topologies lead to an interlacing of SOM nodes due to edge wrapping and more extensive connection distancing between nodes closest to the sampled point.

Rectangular Grid representation due to possible warping of the basic rectangular connectivity. Even a small amount of warping brings nodes located diagonally from each other in the underlying square connectivity, and therefore not directly connected to each other, into close proximity in the modeled space. This results in the two closest nodes to an observation being separated by one intervening node. The toroidal representation produced average topological errors approaching 60% as a result of the wrapping of opposite edges. It may have been expected to see errors of such magnitude for two synthetic data sets, being merely two-dimensional while the torus is effectively three-dimensional, but the degree of error extended to the two literature data sets that were nine- and 197-dimensional as well as the three-dimensional MultiD data set. The edge wrapping effect leads to an interlacing of topologically distant nodes within proximal regions of the data space (Figure 6). While viewed independently, each node still represents a centroid of some region of the data space, but nodes that are direct neighbors are unlikely to be neighbors in the original data space due to the wrapping-induced interlacing.

The growing SOMs fully avoid large TE by effectively adapting to the underlying data during the training process. In both growing maps, the TE was typically below 1−2% and often as low as 0.1%. This has important implications when the map is used to represent large collections of objects (e.g., screenable small molecules) and is queried with a novel object in an effort to identify the most similar subset within the collection. Obviously, the node identified as closest in the SOM to the query observation will hold the most similar data set observations, and extending to adjacent nodes should yield the next most similar set of objects. This will not necessarily be the case with the static maps and, in particular, the toroidal representation and may lead to misinterpretation of the results or a loss of information.

Table 3. Average Connections per Node[a]

	Rectangles	Star	MultiD	BBB	ER
Rectangular Grid	3.6	3.8	3.5	3.2	3.0
Toroid	4.0	4.0	4.0	4.0	4.0
Hexagonal	5.2	5.6	5.0	4.5	4.1
Growing Hex	5.2	5.6	5.6	4.7	4.5
Growing Neural Gas	3.4	3.8	3.9	3.6	3.5

[a] Values presented are the average number of connections per node with trained SOMs. Standard deviations were consistently less than 0.05.

The TE value could be effectively reduced to zero simply by adding connections between every pair of nodes within the SOM. This would clearly eliminate the utility of the TE value as a measure of the SOM's accuracy. It would more importantly render the resulting SOM useless as there would be no way to evaluate which nodes were truly neighbors of those deemed interesting. To examine the complexity of the trained SOMs, we calculated the average number of connections per node (Table 3). As mentioned previously, the static SOMs will have values approaching four and six for the square and hexagonal connectivities, respectively. The variation in actual values seen across data sets is a result of the differing sizes of the SOMs leading to a different number of nodes found on edges and corners, relative to the number of nodes found within the body of the SOM. Not surprisingly, the Growing Hex SOM shows an average number of connections comparable to the nongrowing, hexagonal SOM. Interestingly, the average number of connections for nodes within the Growing Neural Gas SOMs was very similar to those expected for the square connectivities. This result illustrates that the overall complexity of the Growing Neural Gas is quite low, and this further reinforces the conclusion that, once trained, this SOM is much better able to capture the true nature of the modeled data.

The Neural Gas SOM offers the additional benefit of self-modification such that connections between nodes are gained and lost. This is accomplished by reinforcing connections between the two closest nodes to an observed data point and removing links that have not received adequate reinforce-

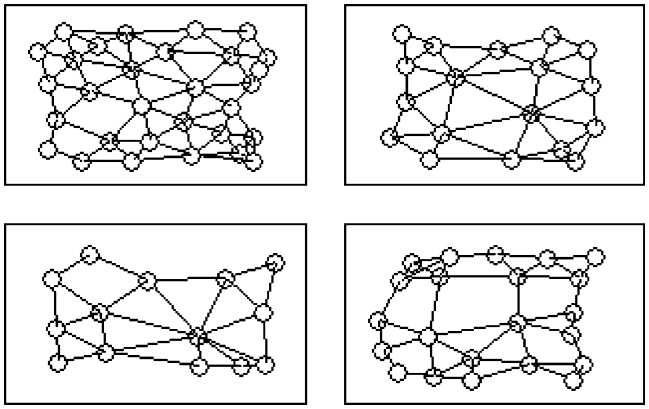

Figure 7. Identification of independent clusters within the Rectangles data set using the Neural Gas SOM. The rectangles illustrate the locations of data points, as described in Figure 4.

ment. The removal of connections is controlled by the a_{max} parameter specifying a maximum connection age. The Rectangles data set provides an interesting test of this self-modifying capacity due to the presence of four disjoint clusters of data. The static maps, although properly parametrized, will inevitably contain nodes within the intercluster spaces (Figure 5, panel B). This results from the unchanging connectivity of the topology and the resulting tug-of-war that occurs for nodes between clusters and serves no purpose for interpretation of the map or in searching for near neighbors. The Neural Gas, however, produces four disjoint graphs, each representing one cluster in the data space (Figure 7), providing clues to the underlying structure of the data. This would be particularly interesting in lower-dimensional visualization of higher-dimensional spaces of real data sets. (For the BBB and ER data sets, this type of cluster segregation was not observed, likely due to the homogeneity of the descriptors used.)

The MultiD data set was developed to provide a more challenging modeling task and consists of one-, two-, and three-dimensional subcomponents embedded within the same three-dimensional space (Figure 1, panel C). The static maps and the Growing Hex topologies were all well able to represent the planar region but exhibited significant twisting of the maps throughout the MultiD data set space. The linear connection between the plane and three-dimensional region proved difficult in that occasional connections were present from the three-dimensional region directly to the plane, and nodes were present in areas lacking data points rather than isolating nodes and connections to the linear region. The Neural Gas isolated all connections between the two- and three-dimensional subspaces to the linear portion and did not exhibit any misplacement of nodes (Figure 8). Furthermore, upon investigation of the nodes and their connections, it was found that those nodes representing the linear region were connected to only two other nodes within the SOM. This surprising result shows that not only was the Neural Gas capable of properly representing the space and the overall topology but the individual nodes were also capable of capturing the true dimensional nature of the modeled data space despite the higher dimensionality of the containing space.

CONCLUSIONS

Self-organizing maps have shown significant promise in numerous fields of application and have become common in cheminformatics and molecular modeling. Improper parametrization, however, can lead to maps that do not correctly represent the modeled space or lead to the misinterpretation of neighborhood relationships. Selection of an appropriate topology and connectivity also has a significant impact on the apparent results when using SOMs. While the static SOMs provide ease of implementation and visualization, growing maps are far more powerful with respect to capturing details of the target data space. To some extent, topological errors can be controlled through parameter choices or more sophisticated techniques such as multiple neighborhood scaling.[23] Parameter choice for static maps, however, does not provide the level of error control exhibited by growing maps. Allowing more extensive flexibility of the SOM topology and connectivity, such as that provided by the Neural Gas algorithm, extends beyond error control. This technique provides the ability to capture details from the modeled data beyond clustering such as lower-dimensional relationships contained within higher-dimensional representations, and thus a more accurate interpretation of results. Ultimately, growing SOMs without predefined connectivity or topology impose the fewest assumptions and the lowest level of constraints on the ability of SOMs to capture details of modeled data spaces.

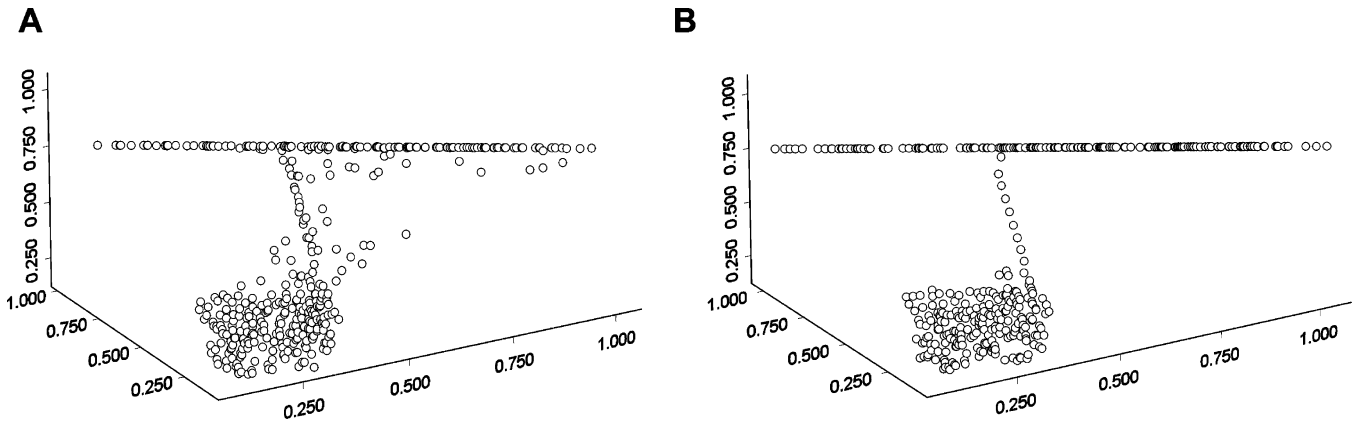

Figure 8. Modeling multiple dimensionalities within a three-dimensional space. (A) Results of training with the Hexagonal Grid. Misplaced nodes are apparent throughout. Connections (omitted for clarity) exist between the various subregions, and twisting of the map is present. (B) Results of training the Neural Gas. No misplaced nodes are apparent, and each subdimensional region is accurately represented. (Training data are shown in Figure 1. These figures are rotated slightly from Figure 1 to emphasize the planar region and the misplaced nodes in A.)

ACKNOWLEDGMENT

The authors gratefully thank Dr. Dan Weaver and Dr. Guy Vigers of Array BioPharma for reviewing the manuscript and providing feedback that greatly improved it. We would also like to thank Dr. Weida Tong of the National Center for Toxicological Research for kindly providing the Estrogen Receptor Binding Dataset (http://www.fda.gov/nctr/science/centers/toxicoinformatics/edkb/index.htm).

Supporting Information Available: The synthetic data sets (Rectangles, Star, and Multi-D) used in this study. This information is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Leo, A. J. Calculating log Poct from structures. *Chem. Rev.* **1993**, *93*, 1281−1306.
(2) DRAGON. http://www.talete.mi.it/dragon_exp.htm (accessed May 2007).
(3) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; Wiley-VCH: Weinheim, Germany, 2000.
(4) Hastie, T.; Tibshirani, R.; Friedman, J. H. *The Elements of Statistical Learning*; Springer: Canada, 2001.
(5) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*, 2nd ed.; Wiley-Interscience: New York, 2000.
(6) Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford University Press: New York, 1996.
(7) Kohonen, T. *Self-Organizing Maps*, 3rd ed.; Springer-Verlag: Berlin, 2006.
(8) Gasteiger, J.; Li, X.; Rudolph, C.; Sadowski, J.; Zupan, J. Representation of molecular electrostatic potentials by topological feature maps. *J. Am. Chem. Soc.* **1994**, *116*, 4608−4620.
(9) Huang, R.; Wallqvist, A.; Covell, D. G. Assessment of in vitro and in vivo activities in the National Cancer Institute's anticancer screen with respect to chemical structure, target specificity, and mechanism of action. **2006**, *49*, 1964−1979.
(10) Rabow, A. A.; Shoemaker, R. H.; Sausville, E. A.; Covell, D. G. Mining the National Cancer Institute's tumor-screening database: identification of compounds with similar cellular activities. *J. Med. Chem.* **2002**, *45*, 818−840.
(11) Ertl, P.; Muhlbacher, J.; Rohde, B.; Selzer, P. Web-based cheminformatics and molecular property prediction tools supporting drug design and development at Novartis. *SAR QSAR Environ. Res.* **2003**, *14*, 321−328.
(12) Selzer, P.; Ertl, P. Applications of self-organizing neural networks in virtual screening and diversity selection. *J. Chem. Inf. Model.* **2006**, *46*, 2319−2323.
(13) von Korff, M.; Hilpert, K. Assessing the predictive power of unsupervised visualization techniques to improve the identification of GPCR-focused compound libraries. *J. Chem. Inf. Model.* **2006**, *46*, 1580−1587.
(14) Noeske, T.; Sasse, B. C.; Stark, H.; Parsons, C. G.; Weil, T.; Schneider, G. Predicting compound selectivity by self-organizing maps: cross-activities of metabotropic glutamate receptor antagonists. *ChemMedChem* **2006**, *1*, 1066−1068.
(15) Otaki, J. M.; Mori, A.; Itoh, Y.; Nakayama, T.; Yamamoto, H. Alignment-free classification of G-protein-coupled receptors using self-organizing maps. *J. Chem. Inf. Model.* **2006**, *46*, 1479−1490.
(16) Ohlson, T.; Aggarwal, V.; Elofsson, A.; MacCallum, R. M. Improved alignment quality by combining evolutionary information, predicted secondary structure and self-organizing maps. *BMC Bioinf.* **2006**, *7*, 357.
(17) Xiao, Y. D.; Clauset, A.; Harris, R.; Bayram, E.; Santago, P., II.; Schmidtt, D. Supervised Self-Organizing Maps in Drug Discovery. 1. Robust Behavior with Overdetermined Data Sets. *J. Chem. Inf. Model.* **2005**, *45*, 1749−1758.
(18) Xiao, Y. D.; Harris, R.; Bayram, E.; Santago Ii, P.; Schmitt, J. D. Supervised Self-Organizing Maps in Drug Discovery. 2. Improvements in Descriptor Selection and Model Validation. *J. Chem. Inf. Model.* **2006**, *46*, 137−144.
(19) Doniger, S.; Hofmann, T.; Yeh, J. Predicting CNS permeability of drug molecules: comparison of neural network and support vector machine algorithms. *J. Comput. Biol.* **2002**, *9*, 849−864.
(20) Fang, H.; Tong, W.; Shi, L. M.; Blair, R.; Perkins, R.; Branham, W.; Hass, B. S.; Xie, Q.; Dial, S. L.; Moland, C. L.; Sheehan, D. M. Structure-activity relationships for a large diverse set of natural, synthetic, and environmental estrogens. *Chem. Res. Tox.* **2001**, *14*, 280−294.
(21) Fritzke, B. *Some competitive learning methods*; Institute for Neural Computation: La Jolla, CA, 1997.
(22) Fritzke, B. A Growing Neural Gas Network Learns Topologies. *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, 1995; pp 625−632.
(23) Murakoshi, K.; Sato, Y. Reducing topological defects in self-organizing maps using multiple scale neighborhood functions. *Biosystems.* **2007**, *90*, 101−104.

CI7001445