JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

Cite This: J. Chem. Inf. Model. 2018, 58, 2528−2535

pubs.acs.org/jcim

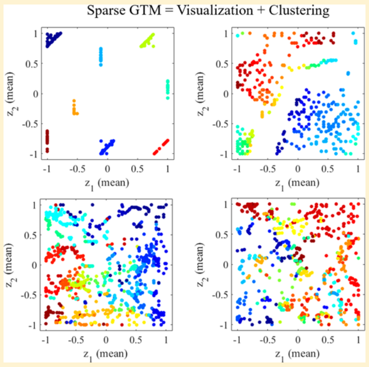
Sparse Generative Topographic Mapping for Both Data Visualization and Clustering

Hiromasa Kaneko*

Department of Applied Chemistry, School of Science and Technology, Meiji University, 1-1-1 Higashi-Mita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

Supporting Information

ABSTRACT: To achieve simultaneous data visualization and clustering, the method of sparse generative topographic mapping (SGTM) is developed by modifying the conventional GTM algorithm. While the weight of each grid point is constant in the original GTM, it becomes a variable in the proposed SGTM, enabling data points to be clustered on two-dimensional maps. The appropriate number of clusters is determined by optimization based on the Bayesian information criterion. Analysis of numerical simulation data sets along with quantitative structure−property relationship and quantitative structure−activity relationship data sets confirmed that the proposed SGTM provides the same degree of visualization performance as the original GTM and clusters data points appropriately. Python and MATLAB codes for the proposed algorithm are available at https://github.com/hkaneko1985/gtm-generativetopographicmapping.



■ INTRODUCTION

In the fields of materials informatics, chemoinformatics, and chemometrics, it is important to visualize data sets. In general, chemical structures are represented by molecular descriptors, spectral data are represented by intensity for each wavelength, and time series data are represented by measured values for each process variable. Such data sets are multidimensional. As it is not possible to confirm the data distributions and proximity relationships of data sets by taking all of their variables into account, the variables are reduced to two or three dimensions. This allows the data sets to be checked visually. Furthermore, by reflecting information on the objective variable *Y*, such as the physical properties and activities of compounds, in the visualization results, it becomes possible to confirm data set information based on *Y* values.

Principal component analysis (PCA)[1] and factor analysis[2] can be used for linear visualization. Kernel PCA,[3] self-organizing maps (SOMs),[4] generative topographic mapping (GTM),[5] locally linear embedding (LLE),[6] isometric mapping (Isomap),[7] and *t*-distributed stochastic neighbor embedding (tSNE)[8] are examples of nonlinear visualization methods. Visualization results of kernel PCA, GTM, LLE, Isomap, and tSNE were compared.[9] It is possible to visualize new data points by constructing visualization models with PCA, FA, SOMs, and GTM; however, LLE, Isomap, and tSNE can be used only to visualize the given data sets, while multidimensional scaling (MDS) has the out-of-sample extension proposed by Trosset and Priebe.[10] Although PCA has no hyperparameter, there exist visualization methods in which hyperparameters must be selected in advance. Examples of hyperparameters include the parameters used in the kernel functions for kernel PCA, map size, number of radial basis functions (RBFs), variance of each RBF, the regularization parameter in an expectation−maximization (EM) algorithm for GTM, the number of neighbor samples considered in Isomap and LLE, and perplexity for tSNE. The hyperparameters in visualization can be optimized so that the *k*-nearest-neighbor-normalized (k3n) error for visualization and reconstruction[9] is minimized since the k3n error approach is based on the standardized errors between the Euclidean distances to neighboring samples before and after projection onto the latent space.

By the use of clustering methods, data points can be divided into groups that have a similar signature based on the similarity between data points. Clustering techniques include hierarchical clustering, such as the group average method and Ward's method,[11] and nonhierarchical clustering or partitional optimization clustering, such as *k*-means clustering[12] and Gaussian mixture models (GMMs)[13] Drawing a dendrogram of the hierarchical clustering allows the state of cluster merging to be checked as a tree. Clustering analyses have been performed to discuss protein−ligand interactions,[14] the diversity of compound libraries,[15] the predictive performance of quantitative structure−activity relationships (QSARs),[16] and scaffold diversity.[17]

Both data visualization and clustering are types of unsupervised learning. No information on the *Y* values of data points (e.g., compounds) is necessary, and only explanatory variables *X* (e.g., molecular descriptors, finger-

Special Issue: Materials Informatics

Received: August 3, 2018
Published: October 23, 2018

prints, spectra, and process variables) are used. For example, in SOMs, clustering can be performed on the basis of the distance to neurons while visualizing the data set. However, SOMs have the following problems:[5]

1 the absence of a cost function

2 the lack of a theoretical background for selecting the learning rate and neighborhood functions to ensure topographic ordering

3 the absence of any general proofs of convergence

4 the fact that the probability density is not defined for SOM models

In addition, data points are not always similar on the SOM, even when they are similar in the original space; similarly, data points are not always similar in the original space, even when they are similar on the SOM.

This study focuses on GTM, which was developed to overcome the above problems with SOMs. GTM is a nonlinear method of reducing the dimension of X variables and transforming data points from the original space to the latent space. Generally, the latent space is two-dimensional to allow the data set to be visualized. Unlike PCA, the coordinates of the two-dimensional map are prepared first, and the data points in the original space are then fitted to the coordinates, changing the shape of the map. A rubberlike sheet (two-dimensional map) is bent, stretched, and shrunk so that it covers every data point. The sheet is then placed on a flat surface, and the data points are projected onto the sheet for visualization. GTM can solve various problems encountered with SOMs.

It takes time to optimize all of the hyperparameters since GTM has a large number of hyperparameters. In GTM, data points that are close to each other in the original space are also close to each other in the two-dimensional map constructed with GTM. GTM has been applied to classification[18] and regression analysis,[19,20] and the GTM algorithm itself has been extended to regression analysis and the inverse analysis of regression models.[21]

Examples of the application of GTM to clustering include the work of Wang and Wu,[22] who carried out gas and oil reservoir clustering and unknown reservoir identification on the basis of the Euclidean distances between data points on the two-dimensional GTM map. Escobar et al.[23] combined GTM and graph theory to cluster process data into normal and abnormal data points on the basis of the similarity between responsibility data points calculated by GTM and applied the results to process monitoring. Bose and Chen[24] developed a new fuzzy clustering method by combining GTM and fuzzy c-means clustering for mixture density distributions.

The above studies achieved data visualization and clustering by combining GTM with some other algorithm. This paper reports a method of sparse generative topographic mapping (SGTM) that modifies the GTM algorithm itself to enable simultaneous data visualization and clustering. In GTM, when the size of the map is $k \times k$, the weights of the normal distribution corresponding to the individual nodes on the map are all $1/k^2$. In the proposed SGTM, the weight $\pi^{(i)}$ for each node i is variable. This has the same meaning as the mixture coefficient of GMMs. In addition to a weight matrix $\mathbf{W}$ and the reciprocal of the variance of the normal distribution β, $\pi^{(i)}$ is also optimized by an EM algorithm. As a result, grid points with $\pi^{(i)} = 0$ do not contribute to the visualization of the data, and the number of normal distributions with $\pi^{(i)} \neq 0$ becomes the number of clusters.

To optimize the number of clusters in SGTM, the Bayesian information criterion (BIC)[25] is calculated for various cluster numbers, and the number of clusters that minimizes the BIC is taken as the optimum. Thus, SGTM simultaneously achieves both data visualization and clustering.

To confirm the visualization and clustering performance of SGTM, two types of numerical simulation data sets are analyzed. SGTM is then applied to quantitative structure—property relationship (QSPR) and QSAR data sets to visualize and cluster compounds.

SPARSE GENERATIVE TOPOGRAPHIC MAPPING (SGTM)

SGTM was developed by modifying GTM. Details of the GTM procedure are given in Appendix A.

The latent space has two dimensions, z_1 and z_2, both running from −1 to 1. In the square enclosed by the four points (−1, −1), (−1, 1), (1, −1), and (1, 1), $k \times k$ grid points (coordinates) are taken to be equally spaced. The ith grid point $\mathbf{z}^{(i)}$ is given as follows:

$$\mathbf{z}^{(i)} = [z_1^{(i)} \quad z_2^{(i)}] \tag{1}$$

The X variables of the jth data point are represented by

$$\mathbf{x}^{(j)} = [x_1^{(j)} \quad x_2^{(j)} \quad \cdots \quad x_m^{(j)}] \tag{2}$$

where m is the number of X variables. Let f be a function that converts $\mathbf{z}^{(i)}$ to $\mathbf{x}^{(j)}$, i.e., $\mathbf{x}^{(j)} = f(\mathbf{z}^{(i)})$, for all i and j. In the original GTM algorithm, $f(\mathbf{z}^{(i)})$ is defined by a linear combination of p RBFs. In terms of neural networks, RBFs are like a middle layer. When the center of the jth RBF $\phi_j(\mathbf{z}^{(i)})$ is $\mathbf{t}^{(j)} = [t_1^{(j)} \ t_2^{(j)}]$, the RBF is expressed as follows:

$$\phi_j(\mathbf{z}^{(i)}) = \exp\left[-\frac{1}{2\sigma^2} \| \mathbf{t}^{(j)} - \mathbf{z}^{(i)} \|^2\right] \tag{3}$$

where σ is the variance of the RBFs. $f(\mathbf{z}^{(i)})$ is given as follows:

$$\begin{aligned} f(\mathbf{z}^{(i)}) &= \phi_1(\mathbf{z}^{(i)})\mathbf{w}_1 + \phi_2(\mathbf{z}^{(i)})\mathbf{w}_2 + \cdots + \phi_p(\mathbf{z}^{(i)})\mathbf{w}_p \\ &= \Phi(\mathbf{z}^{(i)})\mathbf{W} \end{aligned} \tag{4}$$

where the $\mathbf{w}_j$ are the m-dimensional weight vectors, which are going to be found.

The transformation from each grid point in the latent space Z to the original space X has a normal distribution centered on $f(\mathbf{z}^{(i)}) = \Phi(\mathbf{z}^{(i)})\mathbf{W}$, which is given as follows:

$$p(\mathbf{x}|\mathbf{z}^{(i)}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{m/2} \exp\left[-\frac{\beta}{2} \| \Phi(\mathbf{z}^{(i)})\mathbf{W} - \mathbf{x} \|^2\right] \tag{5}$$

where β is the reciprocal of the variance of the normal distribution.

While the weight of each normal distribution in eq 5 is constant at $1/k^2$ in GTM (see eq A.6)), the weight of each normal distribution is variable for SGTM, similar to the case of GMMs. In terms of the a priori distribution on the manifold, a discrete and uniform distribution over the nodes was used in the original GTM, as shown in eq 4 in ref 5. Then the distribution function in data space was given as eq A.6. In the proposed SGTM, the prior distribution is nonuniform, adding some degrees of freedom to the original GTM algorithm, and

DOI: 10.1021/acs.jcim.8b00528
J. Chem. Inf. Model. 2018, 58, 2528−2535

thus, the probability distribution $p(\mathbf{x}|\mathbf{W}, \beta, \boldsymbol{\pi})$ based on all grid points is given as follows:

$$p(\mathbf{x}|\mathbf{W}, \beta, \boldsymbol{\pi}) = \sum_{i=1}^{k^2} \pi^{(i)} p(\mathbf{x}|\mathbf{z}^{(i)}, \mathbf{W}, \beta) \tag{6}$$

where $\pi^{(i)}$ is the weight of the ith normal distribution, $\boldsymbol{\pi}$ denotes the vectors $\pi^{(1)}, \pi^{(2)}, \cdots, \pi^{(k^2)}$, and

$$\sum_{i=1}^{k^2} \pi^{(i)} = 1 \tag{7}$$

$\pi^{(i)}$ has the same meaning as the mixture coefficient for GMMs.

Accordingly, the log-likelihood function in SGTM is given as follows:

$$\begin{aligned} L(\mathbf{W}, \beta, \boldsymbol{\pi}) &= \sum_{i=1}^{n} \ln[p(\mathbf{x}^{(i)}|\mathbf{W}, \beta, \boldsymbol{\pi})] \\ &= \sum_{i=1}^{n} \ln\left[\sum_{j=1}^{k^2} \pi^{(j)} p(\mathbf{x}^{(i)}|\mathbf{z}^{(j)}, \mathbf{W}, \beta)\right] \\ &= \sum_{i=1}^{n} \ln\left\{\left(\frac{\beta}{2\pi}\right)^{m/2} \right. \\ &\qquad \left. \sum_{j=1}^{k^2} \pi^{(j)} \exp\left[-\frac{\beta}{2} \| \Phi(\mathbf{z}^{(j)})\mathbf{W} - \mathbf{x}^{(i)} \|^2\right]\right\} \end{aligned} \tag{8}$$

As well as $\mathbf{W}$ and β, $\pi^{(i)}$ is also determined so as to maximize the likelihood function L. As in GTM, $\mathbf{W}$, β, and $\pi^{(i)}$ are calculated using an EM algorithm.

The responsibility (R) is the conditional probability distribution in the E step of the EM algorithm. This represents the probability that each grid point $\mathbf{z}^{(j)}$ corresponds to a data point $\mathbf{x}^{(i)}$. Since the sum of the $\pi^{(i)}$ is equal to 1 and the $\pi^{(i)}$ refer to the prior probability distributions of the nodes, R is expressed using Bayes' theorem as follows:

$$\begin{aligned} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}, \beta, \boldsymbol{\pi}) &= p(\mathbf{z}^{(j)}|\mathbf{x}^{(i)}, \mathbf{W}, \beta, \boldsymbol{\pi}) \\ &= \frac{\pi^{(j)} p(\mathbf{x}^{(i)}|\mathbf{z}^{(j)}, \mathbf{W}, \beta)}{\sum_{r=1}^{k^2} \pi^{(r)} p(\mathbf{x}^{(i)}|\mathbf{z}^{(r)}, \mathbf{W}, \beta)} \end{aligned} \tag{9}$$

In the ath iteration of the EM algorithm, $\mathbf{W}$, β, and $\boldsymbol{\pi}$ are denoted as $\mathbf{W}_a$, β_a, and $\boldsymbol{\pi}_a$, respectively. The likelihood function for the complete data set, L_{comp}, which is to be maximized in the M step of the EM algorithm, is given as follows:

$$\begin{aligned} L_{\text{comp}}(\mathbf{W}, \beta, \boldsymbol{\pi}) = \sum_{i=1}^{n} \sum_{j=1}^{k^2} & R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a, \boldsymbol{\pi}_a) \\ & \ln[\pi^{(j)} p(\mathbf{x}^{(i)}|\mathbf{z}^{(j)}, \mathbf{W}, \beta)] \end{aligned} \tag{10}$$

As in GTM, the values of $\mathbf{W}_{a+1}$ and β_{a+1} that maximize L_{comp} are given as follows:

$$\sum_{i=1}^{n} \sum_{j=1}^{k^2} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a, \boldsymbol{\pi}_a)[\Phi(\mathbf{z}^{(j)})\mathbf{W}_{a+1} - \mathbf{x}^{(i)}]\Phi^{\mathrm{T}}(\mathbf{z}^{(j)}) = 0 \tag{11}$$

$$\begin{aligned} \frac{1}{\beta_{a+1}} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{k^2} & R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a, \boldsymbol{\pi}_a) \\ & \| \Phi(\mathbf{z}^{(j)})\mathbf{W}_{a+1} - \mathbf{x}^{(i)} \|^2 \end{aligned} \tag{12}$$

The method of Lagrange multipliers is used to maximize L_{comp} under the constraint of eq 7 for $\pi^{(i)}$. Let the function to be maximized be $G(\boldsymbol{\pi}, \lambda)$, where λ is the Lagrange multiplier:

$$G(\boldsymbol{\pi}, \lambda) = L_{\text{comp}}(\mathbf{W}, \beta, \boldsymbol{\pi}) - \lambda\left(\sum_{i=1}^{k^2} \pi^{(i)} - 1\right) \tag{13}$$

The partial derivative of $G(\boldsymbol{\pi}, \lambda)$ with respect to $\pi^{(j)}$ is 0, that is,

$$\frac{\partial G(\boldsymbol{\pi}, \lambda)}{\partial \pi^{(j)}} = \sum_{i=1}^{n} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a) \frac{1}{\pi^{(j)}} - \lambda = 0 \tag{14}$$

Thus,

$$\sum_{i=1}^{n} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a) - \lambda \pi^{(j)} = 0 \tag{15}$$

Summing eq 15 from $j = 1$ to k^2, we obtain

$$\begin{aligned} & \sum_{i=1}^{n} \sum_{j=1}^{k^2} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a) - \lambda \sum_{j=1}^{k^2} \pi^{(j)} = 0 \\ & n - \lambda = 0 \\ & \lambda = n \end{aligned} \tag{16}$$

Thus, from eq 15, the value of $\pi_{a+1}^{(j)}$ that maximizes L_{comp} is as follows:

$$\pi_{a+1}^{(j)} = \frac{1}{n} \sum_{i=1}^{n} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a, \boldsymbol{\pi}_a) \tag{17}$$

where $\mathbf{W}$, β, and $\boldsymbol{\pi}$ are updated iteratively using eqs 9, 11, 12, and 17. In addition, the size of $\mathbf{W}$ is restricted, as in GTM, using the regularization coefficient λ. In SGTM, the size of the map k^2, number of RBFs p, variance of the RBFs σ, and regularization coefficient λ are hyperparameters that must be set beforehand.

Once the SGTM model has been constructed, the probability of existence for each data point $\mathbf{x}^{(i)}$ at each point $\mathbf{z}^{(j)}$ on the two-dimensional map can be calculated as the responsibility using eq 9. The following two methods are used to determine the points on the map from the probability distribution R for each data point:

- **Mean**: weighted average of grid points whose weights are the responsibilitIES (eq 9)
- **Mode**: grid point at which the responsibility (eq 9) is maximized

For clustering, grid points or normal distributions with $\pi^{(i)} = 0$ do not contribute to the visualization of data points, and the number of normal distributions with $\pi^{(i)} \neq 0$ becomes the number of clusters. However, although the number of clusters with $\pi^{(i)} \neq 0$ maximizes the likelihood from the viewpoint of data visualization, it is not optimal and is too high from the viewpoint of clustering. Therefore, a method to optimize the number of clusters based on the BIC is proposed.

To optimize the number of clusters, the BIC is calculated while the number of clusters is decreased one by one. The number of clusters that minimizes the BIC is taken as the optimum. The BIC is calculated as

$$\mathrm{BIC} = -2L(\mathbf{W}, \beta, \boldsymbol{\pi}) + d \cdot \ln(n) \tag{18}$$

where d is the current number of clusters. The flow of the BIC-based cluster number optimization method in SGTM is shown in Figure 1. First, the likelihood is calculated for all data points using the constructed SGTM model. Next, the cluster or grid point with the lowest (nonzero) $\pi^{(i)}$ is selected and denoted as $\pi^{(\min)}$. Before $\pi^{(\min)}$ is changed to zero and the number of clusters is reduced, $\pi^{(\min)}$ is evenly divided and equally assigned to the other nonzero $\pi^{(i)}$. That is, $\pi^{(\min)}/(d-1)$ is added to each nonzero $\pi^{(i)}$ so that the sum of the $\pi^{(i)}$ is equal to 1 as in eq 7. The likelihood and BIC are then recalculated. Repeating this procedure until the number of clusters is equal to 1 gives the BIC values for different numbers of clusters.

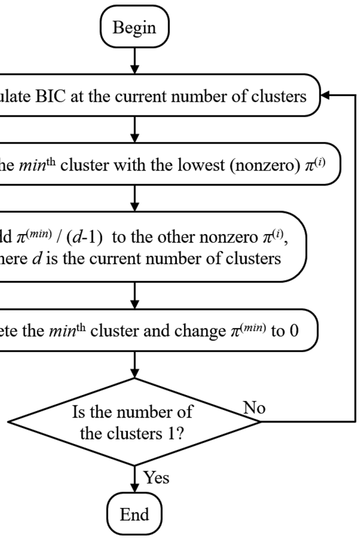

Begin
Calculate BIC at the current number of clusters
Select the min^{th} cluster with the lowest (nonzero) $\pi^{(i)}$
Add $\pi^{(min)}$ / (d-1) to the other nonzero $\pi^{(i)}$, where d is the current number of clusters
Delete the min^{th} cluster and change $\pi^{(min)}$ to 0
Is the number of the clusters 1?
No
Yes
End

Figure 1. Flow of the proposed cluster number optimization method after SGTM modeling.

The number of clusters with the lowest BIC value is then taken as the optimum number of clusters. In addition, the responsibility is calculated for each data point, and the grid point associated with the maximum responsibility becomes the assigned cluster.

Under the process described above, SGTM simultaneously achieves data visualization and clustering. Python and MATLAB codes for SGTM are available at https://github.com/hkaneko1985/gtm-generativetopographicmapping.

■ RESULTS AND DISCUSSION

To verify the effectiveness of the proposed SGTM, numerical simulation data sets as well as QSPR and QSAR data sets were analyzed. Numerical simulation data sets were used to check the clustering performance of SGTM for a data set with multimodal data distributions and to check the visualization and clustering performance of SGTM for a data set in which the data points were continuously distributed and nonlinear. The results for the numerical simulation data sets are shown in the Supporting Information. The candidate hyperparameters for GTM and SGTM are listed in Table 1. The combination of hyperparameter candidates that minimized the k3n error[9] in a grid search was used.

Table 1. Hyperparameter Candidate Values for GTM and SGTM

hyperparameter	candidate values
k of grid points k^2	10, 20, 30, 40, 50
no. of RBFs, $p^{0.5}$	2, 4, ···, 18, 20
variance of each RBF, σ	2^{-5}, 2^{-4}, ···, 2^{2}, 2^{3}
regularization coefficient, λ	0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}

QSPR Study of Aqueous Solubility. A QSPR data set containing 1290 compounds and their aqueous solubilities, expressed as log S (where S is the solubility in moles per liter at a temperature of 20−25 °C[26]), was used to evaluate the performance of SGTM. Sixteen compounds were removed because of duplication. The log S values ranged from −11.62 to 1.58.

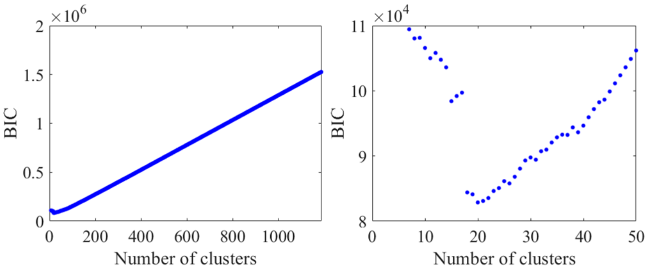
By the use of RDKit,[27] 200 molecular structure descriptors were calculated for the compounds. After removal of descriptors for which the proportion of data points with the same value exceeded 0.8, 10 descriptors were selected using genetic-algorithm-based partial least-squares[28] with 5-fold cross-validation. The selected descriptors were MolWt, HeavyAtomMolWt, MinPartialCharge, FpDensityMorgan2, Chi0v, Chi1, Chi1n, HallKierAlpha, Kappa2, and PEOE_VSA2. These descriptors (but **not** log S) were used as the input variables for both GTM and SGTM.

Table 2 lists the hyperparameter values selected on the basis of the k3n error for the GTM and SGTM methods. With SGTM, although the maximum k candidate value and the minimum σ candidate value were selected, the visualization result did not change significantly as k increased and σ decreased.

Table 2. Optimum Hyperparameter Values Selected on the Basis of k3n Error for the Log S Data Set

hyperparameter	GTM	SGTM
k of grid points k^2	30	50
number of RBFs, $p^{0.5}$	6	6
variance of each RBF, σ	2	2^{-5}
regularization coefficient, λ	10^{-2}	10^{-2}

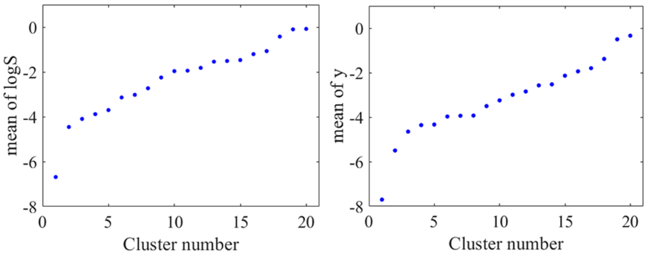
Figure 2 shows the BIC values for various cluster numbers. As the minimum BIC value was attained with 20 clusters, this was set as the optimum number of clusters. Figure 3 shows the average log S value for each cluster. The result was compared with that of k-means clustering in which the number of clusters was the same as that of SGTM. The estimated y value in each



(a) Whole

(b) Only from 0 to 50

Figure 2. BIC value as a function of cluster number for SGTM with the log S data set.

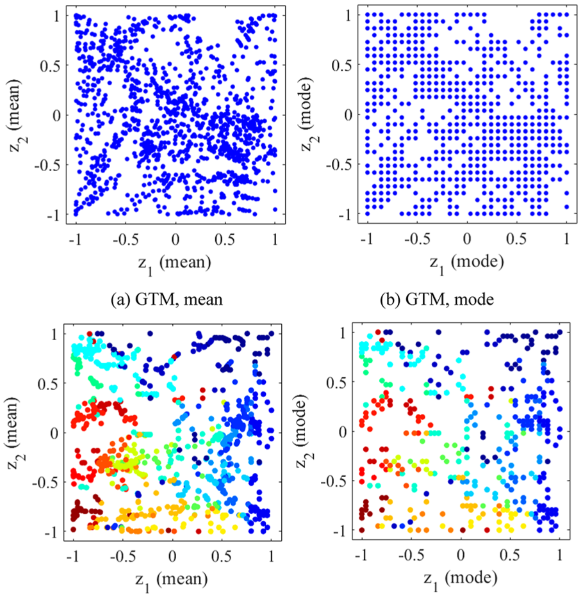


(a) SGTM, RMSE = 1.13 (b) k-means, RMSE = 1.13

Figure 3. Mean of log *S* at each cluster number for SGTM and *k*-means clustering with the log *S* data set. The estimated *y* value in each cluster was set as the average of the measured *y* values of the cluster, and the RMSE was calculated.

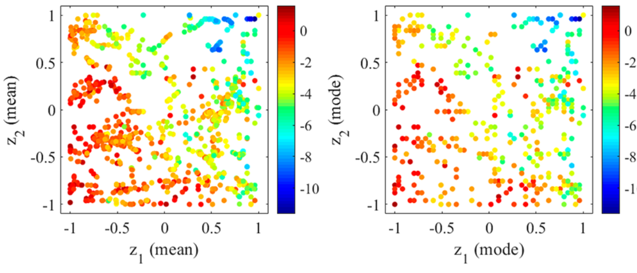
cluster was set as the average of the measured *y* values of the cluster, and the root-mean-square error (RMSE) was calculated. Almost the same results were obtained with *k*-means clustering and the proposed SGTM. It should be noted that since clustering is unsupervised learning, there is no answer for the number of clusters and it is difficult to evaluate the performance of clustering. As shown in Figure 3, the clusters have been rearranged in descending order of the average value of log *S*. It can be seen that some clusters have a low average log *S* and some have a high average log *S*. Although the clustering used only descriptor information, compounds with similar log *S* values have been placed into the same clusters.

Figure 4 shows the visualization results for GTM and SGTM. In each case, the compounds are widely distributed across the two-dimensional map. It is possible to properly visualize compounds using GTM and SGTM. The k3n errors were 1.9 for GTM and 3.0 for SGTM. However, the k3n error of SGTM was sufficiently low that there was no problem with the data visualization. Unlike GTM, SGTM presents cluster information at the same time. From the plots with compounds colored according to their log *S* values (Figure 5), the clustering results of SGTM in Figure 4c,d can be related to the logS values. For example, compounds having high log *S* values are placed into the same clusters. These results confirm that SGTM allows compounds to be visualized and clustered appropriately.



(c) SGTM, mean (d) SGTM, mode

Figure 4. Visualization results for GTM and SGTM with the log *S* data set. Different colors in (c) and (d) denote different clusters.



(a) SGTM, mean (b) SGTM, mode

Figure 5. Visualization results for SGTM with the log *S* data set. The color bar denotes the log *S* value.

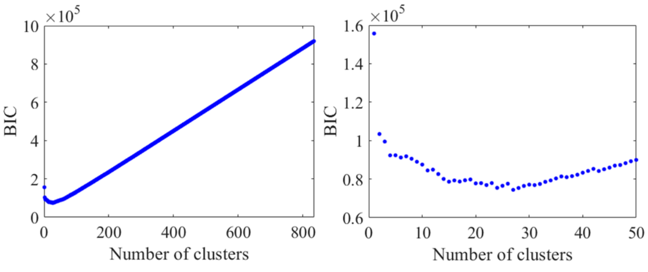
QSAR Study Using pIGC$_{50}$. This data set, downloaded from the Environmental Toxicity Prediction Challenge 2009 website,[29] is an online challenge that invites researchers to predict the toxicity of molecules against *Tetrahymena pyriformis*, expressed as the logarithm of the 50% growth inhibitory concentration in mg/L (pIGC$_{50}$). The data set consists of 1093 compounds. Eight compounds were removed because of duplication. The pIGC$_{50}$ values ranged from −2.67 to 3.34.

In the same way as for the QSPR study, a range of descriptors were calculated, and 10 were selected as explanatory variables. The selected descriptors were MaxAbsEStateIndex, MolWt, HeavyAtomMolWt, MinPartialCharge, Chi0, Kappa2, PEOE_VSA1, PEOE_VSA14, PEOE_VSA8, and PEOE_VSA9.

Table 3 lists the hyperparameter values selected by k3n error for the GTM and SGTM methods. Figure 6 shows the BIC

Table 3. Optimum Hyperparameter Values Selected on the Basis of k3n Error for the Toxicity Data Set

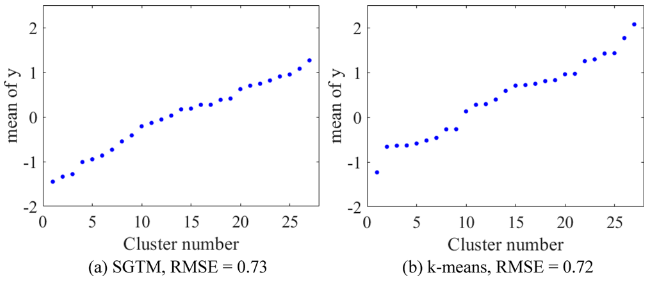
hyperparameter	GTM	SGTM
k of grid points k^2	30	50
number of RBFs, $p^{0.5}$	8	6
variance of each RBF, σ	2^{-3}	2^{-5}
regularization coefficient, λ	10^{-1}	10^{-2}



(a) Whole (b) Only from 0 to 50

Figure 6. BIC value as a function of cluster number for SGTM with the toxicity data set.

values for various cluster numbers given by SGTM. The minimum BIC value occurred with 27 clusters; thus, this was set as the optimum number of clusters. Figure 7 shows the average pIGC$_{50}$ value for each of the 27 clusters using SGTM and k-means clustering in which the number of clusters was the same as that of SGTM. Almost the same results could be obtained with k-means clustering and the proposed SGTM, although there is no answer for the number of clusters and it is difficult to evaluate the performance of clustering since clustering is unsupervised learning. As shown in Figure 7, the clusters were sorted in ascending order of pIGC$_{50}$. From Figure 7, there are clearly clusters with low average pIGC$_{50}$ values and clusters with high average pIGC$_{50}$ values. Thus, the clustering performed using SGTM is appropriate.

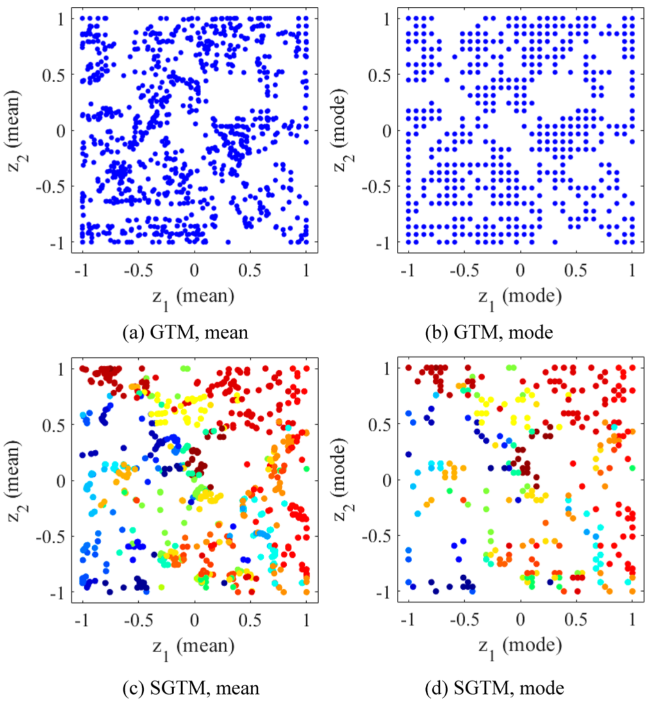


(a) SGTM, RMSE = 0.73

(b) k-means, RMSE = 0.72

Figure 7. Mean of pIGC$_{50}$ at each cluster number for SGTM and k-means clustering with the toxicity data set. The estimated y value in each cluster was set as the average of the measured y values of the cluster, and the RMSE was calculated.

Figure 8 shows the visualization results for GTM and SGTM, respectively. With both GTM and SGTM, the compounds are widely distributed across the two-dimensional maps, and good visualization was achieved. The k3n error was 1.7 for GTM and 3.6 for SGTM. Although higher than that of GTM, the k3n error of SGTM is sufficiently low to obtain visualization results with only slight errors.



(a) GTM, mean

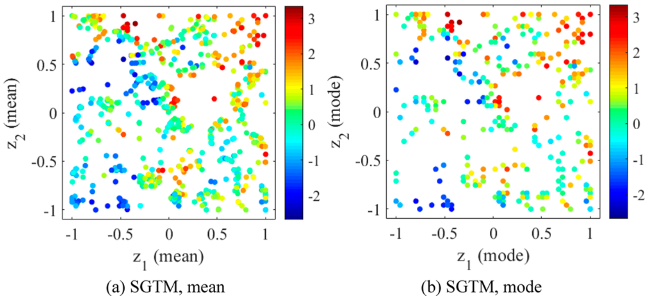
(b) GTM, mode

(c) SGTM, mean

(d) SGTM, mode

Figure 8. Visualization results for GTM and SGTM with the toxicity data set. Different colors in (c) and (d) denote different clusters.

As shown in Figure 8, GTM provides information only about the data distribution on the map. However, SGTM gives information about the data distribution and the clustering on the map. Thus, a larger amount of information can be obtained using SGTM. Figure 9 shows the data distribution of pIGC$_{50}$ given by coloring the compounds with the pIGC$_{50}$ values in the visualization results of SGTM. Comparing Figure 9 with Figure 8c,d shows that compounds with lower pIGC$_{50}$ values are in the same clusters and compounds with higher pIGC$_{50}$ values form separate clusters.



(a) SGTM, mean

(b) SGTM, mode

Figure 9. Visualization results for SGTM with the toxicity data set. The color bar denotes the pIGC$_{50}$ value.

These results confirm that the proposed SGTM simultaneously achieves good visualization and appropriate clustering.

■ CONCLUSION

In this study, to simultaneously achieve data visualization and clustering, a sparse GTM (SGTM) technique was developed by modifying the original GTM algorithm. Whereas the weight of each grid point is a constant $(1/k^2)$ in the original GTM, the proposed SGTM achieves simultaneous clustering with visualization by changing the weight of each grid point, similar to the clustering of GMMs. Furthermore, by optimization of the number of clusters using the BIC value, it is possible to determine the appropriate number of clusters.

Numerical simulation data sets and QSPR/QSAR data sets were analyzed to check the effectiveness of the proposed SGTM. The results confirmed that SGTM provides the same degree of visualization performance as GTM and that SGTM can properly cluster the data points. In the results of the QSPR and QSAR analyses, compounds with similar log S and pIGC$_{50}$ values were placed in the same clusters.

The proposed method enables efficient data analysis because clustering is performed simultaneously when data are visualized in QSPR and QSAR analyses.

■ APPENDIX A. GENERATIVE TOPOGRAPHIC MAPPING (GTM)

Although GTM[5] is basically a data visualization method, it has been extended to regression analysis.[20]

The latent space has two dimensions, z_1 and z_2. Both z_1 and z_2 run from −1 to 1. In the square enclosed by the four points (−1, −1), (−1, 1), (1, −1), and (1, 1), there are $k \times k$ grid points (coordinates). The ith grid point $\mathbf{z}^{(i)}$ is given as follows:

$$\mathbf{z}^{(i)} = [z_1^{(i)} \quad z_2^{(i)}] \tag{A.1}$$

The X variables of the jth data point are represented by

$$\mathbf{x}^{(j)} = [x_1^{(j)} \;\; x_2^{(j)} \;\; \cdots \;\; x_m^{(j)}] \tag{A.2}$$

where m is the number of X variables. Let f be a function that converts $\mathbf{z}^{(i)}$ to $\mathbf{x}^{(j)}$, i.e., $\mathbf{x}^{(j)} = f(\mathbf{z}^{(i)})$, for all i and j. $f(\mathbf{z}^{(i)})$ is defined by a linear combination of p RBFs. In terms of neural networks, RBFs are like a middle layer. When the center of the jth RBF $\phi_j(\mathbf{z}^{(i)})$ is $\mathbf{t}^{(j)} = [t_1^{(j)} \;\; t_2^{(j)}]$, the RBF is expressed as follows:

$$\phi_j(\mathbf{z}^{(i)}) = \exp\left[-\frac{1}{2\sigma^2}\|\mathbf{t}^{(j)} - \mathbf{z}^{(i)}\|^2\right] \tag{A.3}$$

where σ is the variance of the RBFs. $f(\mathbf{z}^{(i)})$ is given as follows:

$$\begin{aligned} f(\mathbf{z}^{(i)}) &= \phi_1(\mathbf{z}^{(i)})\mathbf{w}_1 + \phi_2(\mathbf{z}^{(i)})\mathbf{w}_2 + \cdots + \phi_p(\mathbf{z}^{(i)})\mathbf{w}_p \\ &= \Phi(\mathbf{z}^{(i)})\mathbf{W} \end{aligned} \tag{A.4}$$

where $\mathbf{w}_j$ is the m-dimensional weight vector, which is going to be found.

The transformation from each grid point in the latent space Z to the original space X has a normal distribution centered on $f(\mathbf{z}^{(i)}) = \Phi(\mathbf{z}^{(i)})\mathbf{W}$, which is given as follows:

$$p(\mathbf{x}|\mathbf{z}^{(i)}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{m/2} \exp\left[-\frac{\beta}{2}\|\Phi(\mathbf{z}^{(i)})\mathbf{W} - \mathbf{x}\|^2\right] \tag{A.5}$$

where β is the reciprocal of the variance of the normal distribution. The probability distribution $p(\mathbf{x}|\mathbf{W}, \beta)$ based on all grid points is calculated by adding the probability distributions of eq A.5 for all of the grid points and dividing by the number of grid points k^2:

$$p(\mathbf{x}|\mathbf{W}, \beta) = \frac{1}{k^2}\sum_{i=1}^{k^2} p(\mathbf{x}|\mathbf{z}^{(i)}, \mathbf{W}, \beta) \tag{A.6}$$

Because $p(\mathbf{x}|\mathbf{W}, \beta)$ should express the data distribution of the data points, higher values of the log-likelihood function are preferable. The log-likelihood function is given as follows:

$$\begin{aligned} L(\mathbf{W}, \beta) &= \sum_{i=1}^{n} \ln[p(\mathbf{x}^{(i)}|\mathbf{W}, \beta)] \\ &= \sum_{i=1}^{n} \ln\left[\frac{1}{k^2}\sum_{j=1}^{k^2} p(\mathbf{x}^{(i)}|\mathbf{z}^{(j)}, \mathbf{W}, \beta)\right] \\ &= \sum_{i=1}^{n} \ln\left\{\frac{1}{k^2}\left(\frac{\beta}{2\pi}\right)^{m/2} \right. \\ &\quad \left. \sum_{j=1}^{k^2} \exp\left[-\frac{\beta}{2}\|\Phi(\mathbf{z}^{(j)})\mathbf{W} - \mathbf{x}^{(i)}\|^2\right]\right\}. \end{aligned} \tag{A.7}$$

$\mathbf{W}$ and β are determined so as to maximize the likelihood function L. In GTM, the maximization of L is regarded as an incomplete data problem in which the transformation of $\mathbf{z}^{(i)}$ to $\mathbf{x}^{(j)}$, i.e., $\mathbf{W}$ and β, is unknown. $\mathbf{W}$ and β are calculated using an EM algorithm.

The responsibility (R) is the conditional probability distribution in the E step of the EM algorithm and represents the probability that each grid point $\mathbf{z}^{(j)}$ corresponds to a data point $\mathbf{x}^{(i)}$. R is expressed as follows:

$$R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}, \beta) = p(\mathbf{z}^{(j)}|\mathbf{x}^{(i)}, \mathbf{W}, \beta) = \frac{p(\mathbf{x}^{(i)}|\mathbf{z}^{(j)}, \mathbf{W}, \beta)}{\sum_{r=1}^{k^2} p(\mathbf{x}^{(i)}|\mathbf{z}^{(r)}, \mathbf{W}, \beta)} \tag{A.8}$$

In the ath iteration of the EM algorithm, $\mathbf{W}$ and β are denoted as $\mathbf{W}_a$ and β_a, respectively. The likelihood function for the complete data set, L_{comp}, which is to be maximized in the M step of the EM algorithm, is given as follows:

$$L_{\text{comp}}(\mathbf{W}, \beta) = \sum_{i=1}^{n}\sum_{j=1}^{k^2} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a) \ln[p(\mathbf{x}^{(i)}|\mathbf{z}^{(j)}, \mathbf{W}, \beta)] \tag{A.9}$$

The values of $\mathbf{W}_{a+1}$ and β_{a+1} that maximize L_{comp} are given as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{k^2} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a)[\Phi(\mathbf{z}^{(j)})\mathbf{W}_{a+1} - \mathbf{x}^{(i)}]\Phi^{\mathrm{T}}(\mathbf{z}^{(j)}) = 0 \tag{A.10}$$

$$\frac{1}{\beta_{a+1}} = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{k^2} R_{\mathbf{x}^{(i)},\mathbf{z}^{(j)}}(\mathbf{W}_a, \beta_a)\|\Phi(\mathbf{z}^{(j)})\mathbf{W}_{a+1} - \mathbf{x}^{(i)}\|^2 \tag{A.11}$$

$\mathbf{W}$ and β are updated iteratively using eqs A.8, A.10, and A.11. In addition, the size of $\mathbf{W}$ is restricted as follows:

$$p(\mathbf{W}|\lambda) = \left(\frac{\lambda}{2\pi}\right)^{k^2 m/2} \exp\left\{-\frac{\lambda}{2}\sum_{i=1}^{m}\sum_{j=1}^{k^2} w_j^{(i)}\right\} \tag{A.12}$$

which is the posterior probability distribution of $\mathbf{W}$ given λ, where λ is the regularization coefficient.

In GTM, the size of the map k^2, number of RBFs p, variance of the RBFs σ, and regularization coefficient λ are hyperparameters that must be set beforehand.

Once the GTM model has been constructed, the probability of existence for each data point $\mathbf{x}^{(i)}$ at each point $\mathbf{z}^{(j)}$ on the two-dimensional map can be calculated as the responsibility using eq A.8. The following two methods are used to determine the points on the map from the probability distribution R for each data point:

- **Mean**: weighted average of grid points whose weights are the responsibilities (eq A.8)
- **Mode**: grid point at which the responsibility (eq A.8) is maximized

Python and MATLAB codes for GTM are available at https://github.com/hkaneko1985/gtm-generativetopographicmapping.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.8b00528.

Multimodal simulation data analysis and continuous simulation data analysis (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: hkaneko@meiji.ac.jp.

ORCID

Hiromasa Kaneko: 0000-0001-8367-6476

Notes

The author declares no competing financial interest.

ACKNOWLEDGMENTS

We thank Stuart Jenkinson, Ph.D., from Edanz Group (www.edanzediting.com/ac) for editing a draft of this manuscript.

ABBREVIATIONS

EM, expectation−maximization; GTM, generative topographic mapping; GMM, Gaussian mixture model; LLE, locally linear embedding; MDS, multidimensional scaling; PCA, principal component analysis; QSAR, quantitative structure−activity relationship; QSPR, quantitative structure−property relationship; RBF, radial basis function; SGTM, sparse generative topographic mapping; SOM, self-organizing map; tSNE, *t*-distributed stochastic neighbor embedding; *X*, descriptor or explanatory variable; *Y*, objective variable

REFERENCES

(1) Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37−52.

(2) Akaike, H. Factor Analysis and AIC. *Psychometrika* **1987**, *52*, 317−332.

(3) Muller, K. R.; Mika, S.; Ratsch, G.; Tsuda, K.; Scholkopf, B. An Introduction to Kernel-based Learning Algorithms. *IEEE T. Neural Networ.* **2001**, *12*, 181−201.

(4) Kohonen, T. The Self-organizing Map. *Proc. IEEE* **1990**, *78*, 1464−1480.

(5) Bishop, C. M.; Svensén, M.; Williams, C. K. I. GTM: The Generative Topographic Mapping. *Neural Comput.* **1998**, *10*, 215−234.

(6) Roweis, S. T.; Saul, L. K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323−2326.

(7) Balasubramanian, M.; Schwartz, E. L. The Isomap Algorithm and Topological Stability. *Science* **2002**, *295*, 7a.

(8) van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.

(9) Kaneko, H. K-Nearest Neighbor Normalized Error for Visualization and Reconstruction – A New Measure for Data Visualization Performance. *Chemom. Intell. Lab. Syst.* **2018**, *176*, 22−33.

(10) Trosset, M. W.; Priebe, C. E. The Out-of-sample Problem for Classical Multidimensional Scaling. *Comput. Stat. Data An.* **2008**, *52*, 4635−4642.

(11) Rivera-Borroto, O. M.; Marrero-Ponce, Y.; Garcia-de la Vega, J. M.; Grau-Abalo, R. D. Comparison of Combinatorial Clustering Methods on Pharmacological Data Sets Represented by Machine Learning-Selected Real Molecular Descriptors. *J. Chem. Inf. Model.* **2011**, *51*, 3036−3049.

(12) Bocker, A. Toward an Improved Clustering of Large Data Sets Using Maximum Common Substructures and Topological Fingerprints. *J. Chem. Inf. Model.* **2008**, *48*, 2097−2107.

(13) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer: New York, 2006.

(14) Amari, S.; Aizawa, M.; Zhang, J.; Fukuzawa, K.; Mochizuki, Y.; Iwasawa, Y.; Nakata, K.; Chuman, H.; Nakano, T. VISCANA: Visualized Cluster Analysis of Protein−Ligand Interaction Based on the ab Initio Fragment Molecular Orbital Method for Virtual Ligand Screening. *J. Chem. Inf. Model.* **2006**, *46*, 221−230.

(15) Haigh, J. A.; Pickup, B. T.; Grant, J. A.; Nicholls, A. Small Molecule Shape-Fingerprints. *J. Chem. Inf. Model.* **2005**, *45*, 673−684.

(16) Song, M.; Clark, M. Development and Evaluation of an in Silico Model for hERG Binding. *J. Chem. Inf. Model.* **2006**, *46*, 392−400.

(17) Krier, M.; Bret, G.; Rognan, D. Assessing the Scaffold Diversity of Screening Libraries. *J. Chem. Inf. Model.* **2006**, *46*, 512−524.

(18) Gaspar, H. A.; Marcou, G.; Horvath, D.; Arault, A.; Lozano, S.; Vayer, P.; Varnek, A. Generative Topographic Mapping-Based Classification Models and Their Applicability Domain: Application to the Biopharmaceutics Drug Disposition Classification System (BDDCS). *J. Chem. Inf. Model.* **2013**, *53*, 3318−3325.

(19) Gaspar, H. A.; Baskin, I. I.; Marcou, G.; Horvath, D.; Varnek, A. GTM-Based QSAR Models and Their Applicability Domains. *Mol. Inf.* **2015**, *34*, 348−356.

(20) Gaspar, H. A.; Baskin, I. I.; Marcou, G.; Horvath, D.; Varnek, A. Stargate GTM: Bridging Descriptor and Activity Spaces. *J. Chem. Inf. Model.* **2015**, *55*, 2403−2410.

(21) Kaneko, H. Data Visualization, Regression, Applicability Domains and Inverse Analysis Based on Generative Topographic Mapping. *Mol. Inf.* **2018**, DOI: 10.1002/minf.201800088.

(22) Wang, M.; Wu, Q. Research of Advanced GTM and Its Application to Gas-Oil Reservoir Identification. *Int. J. Patt. Recogn. Artif. Intell.* **2017**, *31*, 1750015.

(23) Escobar, M. S.; Kaneko, H.; Funatsu, K. Combined Generative Topographic Mapping and Graph Theory Unsupervised Approach for Non-linear Fault Identification. *AIChE J.* **2015**, *61*, 1559−1571.

(24) Bose, I.; Chen, X. A Method for Extension of Generative Topographic Mapping for Fuzzy Clustering. *J. Am. Soc. Inf. Sci. Technol.* **2009**, *60*, 363−371.

(25) Fraley, C.; Raftery, A. E. How many clusters? Which clustering method? Answers via model-based cluster analysis. *Comput. J.* **1998**, *41*, 578−588.

(26) Hou, T. J.; Xia, K.; Zhang, W.; Xu, X. J. ADME Evaluation in Drug Discovery. 4. Prediction of Aqueous Solubility Based on Atom Contribution Approach. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 266−275.

(27) http://www.rdkit.org/ (accessed June 21, 2018).

(28) Arcos, M. J.; Ortiz, M. C.; Villahoz, B.; Sarabia, L. A. Genetic-algorithm-based Wavelength Selection in Multicomponent Spectrometric Determinations by PLS: Application on Indomethacin and Acemethacin Mixture. *Anal. Chim. Acta* **1997**, *339*, 63−77.

(29) http://www.cadaster.eu/node/65.html (accessed June 21, 2018).