Using molecular fingerprints and unsupervised learning algorithms to find simulants of chemical warfare agents

Leilei Gu\textsuperscript{1,a}, Xinxin Zhang\textsuperscript{2,b}, Ke Li\textsuperscript{3,c}, Guozhu Jia\textsuperscript{4,d}\textsuperscript{*}

\textsuperscript{1,2,3,4}College of Physics and Electronic Engineering, Sichuan Normal University, Chengdu, China
\textsuperscript{a}sicnu\_guleilei@163.com, \textsuperscript{b}xin13568164982@163.com, \textsuperscript{c}lk\_1275@163.com, \textsuperscript{d}\textsuperscript{*}Jia1689500@126.com

Abstract—The emergence of novel coronavirus highlights the importance of research and development of biological protective materials and functional protective equipment. As an important experimental material, the direct application of chemical warfare agents (CWAs) will cause great pollution to the environment. The effective search for simulants determines the process of CWAs experiments. This paper combines molecular fingerprint and unsupervised learning algorithm to develop a simulants selection framework. A selection strategy is developed based on the silhouette coefficient. The closest simulants are found (GA (TEP/DEEP), GB (DFP), GD (DEHP), HD (CEES), VX (Amiton)) under a threshold (Silhouette coefficient: 0.2). This study can effectively help researchers to find the best approximate simulant to a certain extent.

1. INTRODUCTION
The sudden attack on the novel coronavirus warns of the need to develop biological protection tools ahead of time [1]. Chemical warfare agents (CWAs) are considered an important material for it [2]. The CWAs have obvious toxicity and have related restrictions on laboratory use, and if not effectively tested and disposed of it can cause unpredictable environmental pollution and human hazards [3]. Hence, it is particularly important to find a replacement of the CWAs.

Currently, number of approaches is widely applied for the screening and detection of chemical warfare agents [4-7]. "Simulants" is similar but less toxic chemicals that are very similar to chemical warfare agents, and it is expected that the results of simulants can be correlated with the performance of chemical warfare agents. J. Lavoie et al. found simulants of chemical warfare agents by using chemical informatics [8]. Meanwhile, Charles e. Davidson et al. apply colorimetric sensor arrays to detect chemical warfare agents, and obtained the correct identification rate of CWA of 94% out of the total exposure of 174 independent agent tests [9]. In this paper, we propose a new approach from CWA to find simulant based on molecular fingerprint and unsupervised learning algorithm.

Molecular fingerprints are important carriers of molecular information [10, 11]. It can quickly encode the structure or functional properties of molecules and convert them into binary Numbers (01…01) with adjustable lengths [12]. The values and positions in the vector represent both the structure information and the connection to the atomic level. Many typical studies cleverly use MF as an input variable to develop the QSAR model. The first are a fingerprint-based neural network method proposed by Myint et al., which is used for virtual screening and successfully identifies lead compounds [13]. A bitter-sweet forest platform is established to predict the bitterness and sweetness of compounds...
by combining molecular fingerprints with random forest, and the prediction accuracy reached 95% [14]. Zhong et al. successfully predicted the OH˙ reaction rate constant of organic pollutants based on MF and DNN. This model has similar performance with the QSAR model based on the molecular descriptor [15].

Microarray technology is used to analyze and extract useful information on these massive amounts of data [16]. Clustering algorithm is an unsupervised learning technique widely used in microarray analysis [17]. The most widely used Clustering algorithms in recent years are k-means, Birch and Clustering, etc. [18, 19]. R. Jothi et al. applied k-mean to gene data expression, and the algorithm had advantages of fast convergence, small minimum Mean square error and good clustering quality [16]. Chelsea Parlett-Pelleriti et al. explore age-related meta-memory differences by using modified brier scores and hierarchical clustering [20]. In this paper, we propose a selection framework of CWAs simulant based on molecular fingerprint and clustering algorithm, and then develop the optimal selection strategy. Several typical CWAs are used for experiments.

2. MATERIALS AND METHODS

2.1. CWAs and simulants preparation

Any household or industrial chemicals, if used improperly, may be harmful. A clear distinction has been made for certain chemicals that are particularly toxic and easy to develop, such as CWAs [9]. The type of agent of CWAs is mainly blister agents, blood agents, nerve agents, v, nerve agents, g. Understanding CWAs through the use of less toxic simulants is a common research method, and S.L. bartelt-hunt manually lists several classic CWA simulants [21]. The selection of these mimics is based on the rich experience of scientists to compare the structure and properties of various molecules. This method has achieved great success to a certain extent, but it is very expensive in terms of time and manpower. Therefore, we try to solve these problems by using the current advanced computer technology.

| Serial number | Abbreviation | Chemical compound             | CAS       |
|---------------|--------------|--------------------------------|-----------|
| 0             | HD           | Distilled mustard             | 505-60-2  |
| 1             | CEES         | 2-Chloroethyl ethyl sulfide   | 693-07-2  |
| 2             | CEMS         | 2-Chloroethyl methyl sulfide  | 542-81-4  |
| 3             | CEPS         | Chloroethyl phenyl sulfide    | 5535-49-9 |
| 4             | DEM          | Diethyl malonate              | 105-53-3  |
| 5             | DMA          | Dimethyl adipate              | 627-93-0  |
| 6             | DEA          | Diethyl adipate               | 141-28-6  |
| 7             | MS           | Methyl salicylate             | 119-36-8  |
| 8             | DEP          | Diethyl pimelate              | 2050-20-6 |
Five chemical warfare agents are selected for testing. The number of simulants and chemical compounds in each CWA experiment is different. The chemical information of CWA (HD) and its simulants is shown in Table 1. The information of the remaining all CWAs come from ref. [9].

2.2. Detection framework process
The process of using molecular fingerprints and unsupervised learning algorithms to find simulants of chemical warfare agents is shown in Figure 1. Firstly, collect data onto CWAs and simulants and find the "SMILES" formula that corresponds to them. Secondly, the feature expressions of CWAs and simulants are converted into binary Morgan fingerprint under different bits and radius. Further, select the clustering algorithm, and find the most suitable clustering number (K) to improve the clustering effect by the silhouette coefficient. Finally, the selection strategy is developed based on the silhouette coefficient, and the most similar simulant with CWAs is obtained.

Figure 1. The process of using molecular fingerprints and unsupervised learning algorithms to find simulants of chemical warfare agents

2.3. Molecular fingerprints
The core feature of Morgan's fingerprint is the circular substructure, which is used to calculate the substructure at a certain distance from the atom. Get the "SMILES" string of all molecules by running the "ChemDraw" program using the Python tool. The command "chem.molfromsmiles (SMILES)" in the "RDKit" library creates a "Chem-Mol". "AllChem.GetMorganFingerprintAsBitVect(Chem-Mol, radius, nBits)" converts the string of the corresponding binary fingerprint Morgan. "Radius" is the distance from the atom to the substructure, and "nBits" is the length of the generated vector [22].

2.4. Description of the algorithms

2.4.1. K-means algorithm
K-Means algorithm is a well-known clustering method and it has a wide range of applications [23]. K-means clustering method, as one of the classical algorithms of unsupervised learning, groups data according to the degree of closeness between them and Euclidean distance. The main parts are described below. Suppose there is a data set \( X, X=\{x_1, \ldots, x_N\}, x_N \in R_d \). The M-clustering problem optimizes the clustering criterion by dividing the data set into M disjoint subsets (Clusters) \( C_1, \ldots, C_M \). The most commonly used clustering criterion is the sum of the squares of the euclidean distances between each data point \( x_i \) and the centroid \( m_k \) (Cluster center) of the subset \( C_k \) containing \( x_i \). This criterion is called clustering error, which mainly depends on the clustering center \( m_1, \ldots, m_M \).
\[ E(m_1, ..., m_d) = \sum_{i=1}^{N} \sum_{k=1}^{M} I(x_i \in C_k) \| x_i - m_k \|^2 \]  

(1)

If \( X \) is true, then \( I(X) = 1 \), otherwise 0. The experimental steps are as follows: (1) Randomly select K centers; (2) Walk through all the samples and divide them into the nearest center; (3) There are K clusters after partition, and the average value of each cluster are calculated as the new center of mass; (4) Repeat step (2) Until the stop condition is reached. Stop condition: the clustering center will no longer change. All distances are the smallest. The number of iterations reaches the set value. The cost function is the sum of squared errors (SSE).

2.4.2. Birch algorithm

Birch is short for balanced iterative protocol and hierarchical approach to clustering. Its cluster only needs to scan the data set once. The parameter is the number of selectable classes of birch. The number of classes is specified by the user, and the clustering result will return the corresponding number of clusters [24]. In this experiment, a Tree structure is used to help it quickly cluster. This number structure is similar to the balanced B+ tree, which is generally called clustering feature Tree (CF tree). Each node (Including leaf nodes) of the tree is composed of several clustering features (CF). The CF of the internal node has a pointer to the leaf node, which are linked with a two-way linked list.

2.4.3. Spectral Clustering algorithm

Spectral clustering is a clustering method based on graph theory [25]. It can identify an arbitrary shape of sample space and converge on the global most solvable. The basic idea is to use the similarity matrix of sample data to perform feature decomposition to obtain feature vectors for clustering class. It has nothing to do with the characteristics of samples but only the number of samples. Basic idea: consider the samples as vertices, and the similarity between samples as weighted edges, thus turning the clustering problem with a graph segmentation problem: find a method of graph segmentation to make the weight of the edges connecting different groups as low as possible (The similarity between groups should be as low as possible), and the weight of the edges to the group should be as high as possible (The similarity within the group should be as high as possible).

2.4.4. Gaussian Mixture Model

Gaussian Mixture Model (GMM), is a clustering algorithm widely used in the industry, the method uses gaussian distribution as a parameter model, and uses expectation maximization (EM) algorithm for training [26]. The final attribution class is determined by the probability of belonging to a certain class. Basic idea: The probability distribution of any shape can be approximated by multiple Gaussian distribution functions. The GMM is composed of multiple single gaussian density distributions (Gaussian). Each gaussian is called a "Component" and together they form the probability density function of GMM.

2.4.5. Mini Batch KMeans algorithm

Mini Batch KMeans is a can keep the clustering accuracy but can greatly reduce the computing time of clustering model, using data from a small Batch subset to reduce computing time, is still trying to optimize the objective function at the same time, this so-called mini batch is to point to each training algorithm of random subset of data, using the random data for training, greatly reduce the computing time, reduce the K-Means algorithm convergence time, but slightly worse than the standard algorithm.

2.4.6. Hierarchical Clustering algorithm

Hierarchical clustering is a method of hierarchical classification until a certain condition is met [27]. There are two main types: (1) Cohesion from bottom to top. Each object is first treated as a cluster, and then the clusters are merged into larger and larger clusters until all objects are in one cluster, or some termination condition is satisfied split; (2) Form top to bottom. All objects are first placed in the same cluster, and then subdivided into smaller and smaller clusters until each object forms its own cluster or
reaches some termination condition (Rarely used). Algorithm steps: (1) Classify each object into one class, and N classes are obtained in total. Each class contains only one object; (2) Find the closest two classes and merge them into one class, so the total number of classes is one less; (3) Recalculate the distance between the new class and all the old classes; (4) Repeat steps 2 and 3 until you finally merge into one class (Which contains N objects).

2.5. Silhouette coefficient
The selection of clustering number (K) is very important, which can directly affect the final clustering effect. In order to find a suitable K value, the silhouette coefficient (Referred to as SC in this experiment) is used for exploration. It combines the Cohesion and Separation of clustering, to evaluate the effect of model. The value is between -1 and 1, and the larger the value, the better the clustering performance [28].

The specific calculation approach is as follows:
(1) For the $i$ th element $X_i$, the average distance between $X_i$ and all other elements in the same cluster is calculated, denoted as $A_i$, which is used to quantify the degree of agglomeration in the cluster;
(2) Select a cluster $b$ outside $X_i$, calculate the average distance between all points in $X_i$ and $b$, traverse all other clusters, and find the closest average distance, denoted as $B_i$, to quantify the degree of separation between clusters; (3) For element $X_i$, the $CS_i = (B_i - A_i) / \max(A_i, B_i)$; (4) The all silhouette coefficients are calculated, and the average coefficient is the overall value of the current cluster.

3. RESULTS AND DISCUSSION
At different fingerprint size (50, 100, 200, 500, 1000) and radius (1, 2), molecular fingerprints are first used to convert “SMILES” of CWA and its imitations of binary vectors. After obtaining all the material characteristics, the silhouette coefficient (The range of K values is $2 \sim (\text{The number of datasets}-1)$) under different classes is continuously iterated to find the most suitable K value of clustering.

The simulant clustering results of HD is shown in Table 4 (Appendix). The silhouette coefficients under different characteristics are different, and the coefficients under the same characteristics are also different. Among them, of all the highest scores, the spectral clustering shows the worst performance, and the highest coefficient value is less than 0.250 when the radius is 1 and the number of fingerprint size is 50. Meanwhile, when the radius is 1 and the number of bits is 1000, the highest values corresponding to the other five models are all 0.430. The larger the number of bits, the more molecular information is extracted, but it is also likely to bring dimension disaster to unsupervised algorithms. The size of the silhouette coefficients is extremely important to this point. The high and low scores imply the performance of the model, which can reflect a convincing force for the clustering results, and thus provide a basis of the selection of simulant objects to a certain extent.

The distribution of SC values for GA, GB, GD and VX under different radius and different fingerprint size is shown in Appendix 1. The scatter plot of GA, GB, GD and HD shows that the best SC of spectral clustering is the lowest compared to several other algorithms. This indicates that the graph theory-based calculation method is not suitable for this data feature. This may be related to data having binary features and data set size being too low.
Further, the corresponding optimal K values under each set of features and algorithms are clustered as the classes. All simulants in the same class as CWAs are extracted as the most likely targets, as shown in Appendix Table 1 ~ Table 5.

For GA, the results obtained under different conditions and algorithms vary greatly (See Appendix Table 1). Combined with the distribution of its silhouette coefficient, when the radius is 2 and the number of bits is 200, the coefficient corresponding to the k-means algorithm is the largest. The clustering result is 1, 2, 6. Hence, DEHP, DEEP, and TEHP are considered the most likely alternatives to GA.

For the clustering of GB, the results are shown in Appendix Table 2. The distribution shows an interesting phenomenon. Of all the suspected simulants, the serial number 7(DFP) showed better results (Except that the Gaussian Mixture Model yielded no results when the radius is 1 and the number of fingerprint size is 100, 200, 500, and 1000). This is similar to the simulant ordering results calculated by tanimoto coefficient (TC), and the value is 0.875 [9].

For GD, its results are shown in Appendix Table 3. The result distribution difference is obvious. The best simulants corresponding to Birch and GMM at radius 1 are different from other cases. Their intersection is 14 (DPCP) and 15 (DOP). Meanwhile, the intersection set of 3 (DFP) is taken in the remaining cases, which is consistent with the result obtained by the TC value. The Birch and Hierarchical algorithm with radius of 1 and bits of 200 is optimal (0.202) considering the silhouette coefficient. The coefficient values are still low, indicating that the results at this time need to be further explored.

The data set corresponding to HD is shown in Appendix Table 4. Take the intersection of all the results, and the serial number 1 (Corresponding to CEES) are left. In addition, 2 (CEMS) and 3 (CEPS) are also considered as substitutes for CWAs in most cases. The error can be measured by the silhouette coefficient.

For VX, its clustering results (See Appendix Table 5) show that the simulant with sequence number 1 (Amiton) can be better used as a substitute for VX. This is consistent with the Euclidean distance. However, some binary features and algorithms still cannot find the best simulant that can match VX. Their clustering conditions may be more demanding.

Based on the above results, we find that after Morgan fingerprints are effectively extracted from the molecular structure/function of CWAs and simulants, the clustering effects of different unsupervised learning algorithms are measured using silhouette coefficients, which will be used as the confidence
that the results can be adopted. Hence, an effective simulant sorting model is proposed. The details are as follows:

Using the silhouette coefficient as the weight, according to the above results, we can calculate the sum of all the simulants weight values, and then as the final score. The calculation formula is shown in Eq. (2). In order to exclude the case of too small coefficient, a threshold value is set to \( T \) for filtering. \( SC_n \) is the silhouette coefficient of the simulant obtained under each condition. Considering the clustering of five CWAs, the experiment is set to 0.2.

\[
Final score = \begin{cases} 
\sum SC_n & SC \geq T \\
0 & SC < T 
\end{cases}
\]  

(2)

Final calculated scores of the five CWAs are shown in Figure 3. It can be seen from the graph that under this dataset, the simulant of the alternative GA is 6 (TEP) and 2 (DEEP), GB is 7 (DFP), GD is 4 (DEHP), HD is 1 (CEES), and the VX is 1 (Amiton). In particular, the GD has the lowest score of the optimal simulant, which is only 0.404. When the threshold is raised above 0.202, it will not find the simulant. In order to improve this effect, data set expansion and algorithm optimization will be important work in the future.

Figure 3. The final score distribution of 5 CWAs simulants (See Table 1 and Appendix for the simulants corresponding to the sequence number)

4. CONCLUSION

In this study, we propose a CWAs simulant detection framework based on molecular fingerprints and unsupervised learning algorithms. It could help biochemists find the most appropriate simulants faster and more efficiently at the lowest environmental cost. Through experiments on five chemical warfare agents, the closest simulants are found (GA (TEP/DEEP), GB (DFP), GD (DEHP), HD (CEES), VX (Amiton)) under a certain threshold (silhouette coefficient: 0.2). This research has certain significance of the extraction and development of biological protection materials in the recent global outbreak of public health events. Next work, the number and type of CWAs will be increased, and the simulants data set will be expanded. Meanwhile, high-performance devices can be used to extract more fingerprints, and advanced clustering algorithms will improve the frameworks.

ACKNOWLEDGEMENT

This work has been supported by the Major Cultivation Project of Education Department in Sichuan Province, China (18CZ0006) and the Training Program of Innovation and Entrepreneurship for Undergraduates in Sichuan Province, China (No. S202010636015).
Appendix

S1. K value distribution under different clustering algorithms in GA simulation object clustering (a: K-means, b: Birch, c: Spectral, d: Gaussian Mixture Model, e: Hierarchical, f: Mini-Batch-KMeans)
S2. K value distribution under different clustering algorithms in GB simulation object clustering (a: K-means, b: Birch, c: Spectral, d: Gaussian Mixture Model, e: Hierarchical, f: Mini-Batch-KMeans)

S3. K value distribution under different clustering algorithms in GD simulation object clustering (a: K-means, b: Birch, c: Spectral, d: Gaussian Mixture Model, e: Hierarchical, f: Mini-Batch-KMeans)
S4. K value distribution under different clustering algorithms in VX simulation object clustering (a: K-means, b: Birch, c: Spectral, d: Gaussian Mixture Model, e: Hierarchical, f: Mini-Batch-KMeans)

Table 1 Simulant results of GA found by different algorithms under different fingerprints (number of fingerprint size and radius)

| Fingerprint size | K-means | Birch | Spectral |
|------------------|---------|-------|----------|
| 50               | 1, 2, 6, 13 | 2, 6, 13 | 13 | 2, 6, 13, 14 | 13 | 2, 6, 13 |
| 100              | 1, 2, 6, 13 | 1, 2, 6 | 1, 10, 13, 14, 15 | ~ | ~ | 1, 2, 6 |
| 200              | 1, 2, 6, 13 | 1, 2, 6 | 1, 8, 10, 12~15 | ~ | ~ | 2, 3, 5, 6, 13 | 1, 2, 6 |
| 500              | ~ | ~ | 10 | ~ | ~ | ~ |
| 1000             | ~ | ~ | 10, 13, 14, 15 | ~ | 1, 2, 6, 9, 11, 13 | ~ |

| Fingerprint size | Mini-Batch-Kmeans | GMM | Hierarchical |
|------------------|-------------------|-----|--------------|
| 50               | 2, 6, 13 | 2, 6, 13 | 10, 13, 14, 15 | ~ | 6 | 2, 6, 9, 11, 13 |
| 100              | 1, 2, 6, 13 | 1, 2, 6 | 1, 10, 13, 14, 15 | 2, 6, 13 | 2, 6, 13 | 1, 2, 6 |
| 200              | 2~7, 13, 14 | 2, 6 | 1, 10, 13, 14, 15 | 2, 6, 13 | ~ | 2, 6 |
| 500              | 1, 2, 6, 8~13, 15 | ~ | 1, 10, 13, 14, 15 | 2, 6 | ~ | ~ |
| 1000             | ~ | ~ | 1, 10, 13, 14, 15 | 2, 6 | ~ | ~ |
Table 2 Simulant results of GB found by different algorithms under different fingerprints (number of fingerprint size and radius)

| Fingerprint size | K-means | Birch | Spectral | Mini-Batch-Kmeans | GMM | Hierarchical |
|------------------|---------|-------|----------|--------------------|-----|--------------|
| 50               | 1       | 2     | 1        | 2                  | 1   | 2            |
|                  | 7       | 7     | 7        | 7                  | 7   | 7            |
| 100              | 2       | 1     | 2        | 1                  | 2   | 1            |
|                  | 3       | 7     | 7        | 7                  | 1   | 2            |
| 200              | 1       | 2     | 1        | 2                  | 1   | 2            |
|                  | 7       | 7     | 7        | 7                  | 7   | 7            |
| 500              | 3       | 7     | 7        | 7                  | 7   | 7            |
|                  | 1       | 2     | 1        | 2                  | 1   | 2            |
| 1000             | 4       | 1     | 3        | 6, 8               | 4   | 1, 3, 6, 8   |

Table 3 Simulant results of GD found by different algorithms under different fingerprints (number of fingerprint size and radius)

| Fingerprint size | K-means | Birch | Spectral |
|------------------|---------|-------|----------|
| 50               | ~       | 1, 3  | 14, 15   |
|                  | 3       | 1, 3  | 1, 3     |
| 100              | 1, 3    | 1, 3  | 1, 3     |
|                  | 6, 8    | 1, 3  | 1, 3     |
| 200              | 1, 3    | 3     | 4, 13, 14, 15 |
|                  | 1, 3    | 1, 3  | 1, 3     |
| 500              | 1, 3    | 1, 3  | 1, 3     |
|                  | 13, 14, 15 | 1, 3 | 1, 3 |
| 1000             | ~       | 1, 3  | 13, 14, 15 |
|                  | 1, 3    | 1, 3  | 1, 3     |
|                  | 13, 14, 15 | 1, 3 | 1, 3 |

| Fingerprint size | Mini-Batch-Kmeans | GMM | Hierarchical |
|------------------|-------------------|-----|--------------|
| 50               | 1, 3, 6, 8, 10, 12, 14 | 4, 10, 13, 14, 15 | 1, 3, 6, 8 |
|                  | 1, 3, 6, 8         | 4, 10, 13, 14, 15 | 3, 1, 3, 6, 8 |
| 100              | 1, 3, 6, 8, 14     | 4, 10, 13, 14, 15 | ~       | 3, 1, 3, 6, 8 |
| 200              | 1, 3, 6, 8         | 4, 10, 13, 14, 15 | ~       | 1, 6, 8 |
| 500              | 1, 3, 6, 8         | 4, 10, 13, 14, 15 | 1, 3, 6, 8 | ~       |
| 1000             | ~                  | 4, 10, 13, 14, 15 | ~       | ~       |

Table 4 Simulant results of HD found by different algorithms under different fingerprints (number of fingerprint size and radius)

| Fingerprint size | K-means | Birch | Spectral |
|------------------|---------|-------|----------|
| 50               | 1, 2    | 3     | 1, 2, 3  |
|                  | 1       | 2     | 1, 2, 3  |
| 100              | 1, 2    | 3     | 1, 2, 3  |
|                  | 1       | 2     | 1, 2, 3  |
| 200              | 1, 2    | 3     | 1, 2, 3  |
|                  | 1~6, 7  | 1, 2  | 1, 2, 3  |
| 500              | 1, 2    | 3     | 1, 2, 3  |
|                  | 1~6, 7  | 1, 2  | 1, 2, 3  |
| 1000             | 1, 2    | 3     | 1, 2, 3  |
|                  | 1~6, 7  | 1, 2  | 1, 2, 3  |

| Fingerprint size | Mini-Batch-Kmeans | GMM | Hierarchical |
|------------------|-------------------|-----|--------------|
| 50               | 1, 2              | 1, 3| 1, 2, 3 |
|                  | 1, 2              | 1, 3| 1, 2, 3 |
| 100              | 1, 2              | 1, 3| 1, 2, 3 |
|                  | 1, 2              | 1, 3| 1, 2, 3 |
| 200              | 1, 2              | 1, 3| 1, 2, 3 |
|                  | 1~6, 7            | 1, 2| 1, 2, 3 |
| 500              | 1, 2              | 1, 3| 1, 2, 3 |
|                  | 1~6, 7            | 1, 2| 1, 2, 3 |
| 1000             | 1, 2              | 1, 3| 1, 2, 3 |
|                  | 1~6, 7            | 1, 2| 1, 2, 3 |
Table 5: Simulant results of VX found by different algorithms under different fingerprints (number of fingerprint size and radius)

| Fingerprint size | K-means | Birch | Spectral | Mini-Batch -Kmeans | GMM | Hierarchical |
|------------------|---------|-------|----------|--------------------|-----|--------------|
| 50               | 1       | 1     | 1        | 1                  | 1   | 1            |
|                  | ~       |       |          |                    |     |              |
| 100              | 1       | 1     | ~        | 1                  | 1   | ~            |
|                  |         |       |          |                    |     |              |
| 200              | 1       | 1     | 1        | 1                  | 1   | 1            |
|                  | ~       |       |          |                    |     |              |
| 500              | 1       | 1     | 1        | ~                  | 1   | ~            |
|                  |         |       |          |                    |     |              |
| 1000             | 1       | 1     | 1        | 1                  | 1   | 1            |

REFERENCES

[1] Li F, Feng ZC, Shi Y. Proposal for prevention and control of the 2019 novel coronavirus disease in newborn infants. Archives of Disease in Childhood-Fetal and Neonatal Edition. 2020.

[2] Truong Q, Wilusz E. Advances in chemical and biological protective clothing. Smart Textiles for Protection: Elsevier; 2013. p. 364-77.

[3] Glasby G. Disposal of chemical weapons in the Baltic Sea. Science of the total environment. 1997;206:267-73.

[4] Sayago I, Matatagui D, Fernández MJ, Fontecha JL, Jurewicz I, Garriga R, et al. Graphene oxide as sensitive layer in Love-wave surface acoustic wave sensors for the detection of chemical warfare agent simulants. Talanta. 2016;148:393-400.

[5] Yu H-G, Lee J-H, Park D-J, Nam H-W, Park B-H. Efficient detection algorithm of chemical warfare agents for FTIR-based hyperspectral imagery using SVM classifier. Imaging Spectrometry XXII: Applications, Sensors, and Processing: International Society for Optics and Photonics; 2018. p. 107680H.

[6] Jeong BG, Park DY, Yang K, An S-J, Park C, Lee C, et al. Detection of Chemical Warfare Agent Simulants by using Fluorescence Modulation of Rhodamine 6G/Ag Nanowires. Journal of the Korean Physical Society. 2019;75:267-31.

[7] L’Hermite D, Vors E, Vercouter T, Moutiers G. Evaluation of the efficacy of a portable LIBS system for detection of CWA on surfaces. Environmental Science and Pollution Research. 2016;23:8219-26.

[8] Lee D. Application of Artificial Neural Network to Prompt Gamma Neutron Activation Analysis for Chemical Warfare Agents Identification. Idaho National Lab.(INL), Idaho Falls, ID (United States); 2019.

[9] Lavoie J, Srinivasan S, Nagarajan R. Using cheminformatics to find simulants for chemical warfare agents. Journal of hazardous materials. 2011;194:85-91.

[10] Davidson C, Dixon MM, Williams BR, Kilper GK, Lim SH, Martino RA, et al. Detection of Chemical Warfare Agents by Colorimetric Sensor Arrays. ACS sensors. 2020.

[11] Tabei Y. Scalable Prediction of Compound-protein Interaction on Compressed Molecular Fingerprints. Molecular Informatics. 2020.

[12] Huang Y-A, You Z-H, Chen X. A systematic prediction of drug-target interactions using molecular fingerprints and protein sequences. Current Protein and Peptide Science. 2018;19:468-78.

[13] Glen RC, Bender A, Amby CH, Carlsson L, Boyer S, Smith J. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. IDrugs. 2006;9:199.

[14] Myint K-Z, Wang L, Tong Q, Xie X-Q. Molecular fingerprint-based artificial neural networks QSAR for ligand biological activity predictions. Molecular pharmaceutics. 2012;9:2912-23.

[15] Banerjee P, Preissner R. BitterSweetForest: a random forest based binary classifier to predict bitterness and sweetness of chemical compounds. Frontiers in chemistry. 2018;6:93.
[16] Zhong S, Hu J, Fan X, Yu X, Zhang H. A deep neural network combined with molecular fingerprints (DNN-MF) to develop predictive models for hydroxyl radical rate constants of water contaminants. Journal of hazardous materials. 2020;383:121141.

[17] Jothi R, Mohanty SK, Ojha A. DK-means: a deterministic k-means clustering algorithm for gene expression analysis. Pattern Analysis and Applications. 2019;22:649-67.

[18] Jiang D, Tang C, Zhang A. Cluster analysis for gene expression data: a survey. IEEE Transactions on knowledge and data engineering. 2004;16:1370-86.

[19] Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. Pattern recognition. 2003;36:451-61.

[20] Johnson SC. Hierarchical clustering schemes. Psychometrika. 1967;32:241-54.

[21] Bartelt-Hunt SL, Knappe DR, Barlaz MA. A review of chemical warfare agent simulants for the study of environmental behavior. Critical reviews in environmental science and technology. 2008;38:112-36.

[22] Cortes-Ciriano I. Bioalerts: a python library for the derivation of structural alerts from bioactivity and toxicity data sets. Journal of cheminformatics. 2016;8:13.

[23] Yang B, Fu X, Sidiropoulos ND, Hong M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. Proceedings of the 34th International Conference on Machine Learning-Volume 70: JMLR. org; 2017. p. 3861-70.

[24] Wu Y. Parameter Free Clustering Algorithm Based on Density and Natural Nearest Neighbor. 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM): IEEE; 2019. p. 115-9.

[25] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems2002. p. 849-56.

[26] Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. 2018.

[27] Bongiorno C, Miccichè S, Mantegna RN. Nested partitions from hierarchical clustering statistical validation. arXiv preprint arXiv:190606908. 2019.

[28] Aranganayagi S, Thangavel K. Clustering categorical data using silhouette coefficient as a relocating measure. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007): IEEE; 2007. p. 13-7.