A New Clustering Algorithm Based on Pattern Extraction in Molecular Fingerprints

Bernardo Palacios Bejarano, Gonzalo Cerruela García, Irene Luque Ruiz, Nicolás García Pedrajas, Miguel Ángel Gómez-Nieto
University of Córdoba. Department of Computing and Numerical Analysis
Campus de Rabanales. Albert Einstein Building. E-14071
Córdoba, Spain
{i82pabeb, gcerruela, iluque, npedrajas, mangel}@uco.es

Abstract— **In this paper an algorithm for the extraction of patterns in chemical fingerprints is described. As input this algorithm uses a fingerprint representation of the molecule dataset, generating a group of consistent disjoint patterns also represented as binary arrays, which are satisfied by not necessarily disjoint subsets of molecules in the dataset. The algorithm has been completely developed in Java, allowing its integration into free applications of computational chemistry. The algorithm has been tested, and the use of the patterns instead of the original fingerprints has presented an increase in the efficiency in the processes of datasets classification. The results show that it is possible to reconstruct the original fingerprints using the final group of patterns that characterize all the elements of the dataset.**

Keywords- clustering algorithms, chemical fingerprint, molecular classification

I. INTRODUCTION

Clustering methods are widely used in the study of molecular and biological science and in the pharmaceutical industry to group molecules in order to interpret chemotypes present in the dataset, and as a pre-processing stage in the high-throughput screening (HTS) process.

An adequate representation of the structural and physicochemical features of chemical compounds is crucial for the application of a specific clustering method. In recent years different representation models have been proposed, but the use of graph and arrays are the most employed [1, 2]. In a graph (molecular graph) the molecules are represented by means of nodes and the relationships between the atoms (using the graph edges), being necessary to maintain information of the nature and characteristic of atoms and links. The advantage of this representation is that it can be easily projected on different matrix structures, totally or partially representing information from the molecular graph, and these matrix structures are very efficient for developing algorithms useful in Computational Chemistry.

The molecular structure vector representation (fingerprints) [3] gives a more efficient algorithmic behaviour [4]. They consist of binary arrays that are built using the information from the molecular graph, storing information about structural elements of the chemical species. Different fingerprint models have been proposed based on different molecular graph projections. Although in the fingerprint construction process not all the information of the molecular graph can be stored, these structures have proved to be extremely efficient in many investigations in Computational Chemistry [1-5].

The use of molecular graphs to measure the similarity between molecules results in a high computational cost, although the measuring of similarity between fingerprints is very quick and efficient process and these structures are more used in many of the classification models, screening and QSPR / QSAR applications [6-9].

Many classification and screening models for chemical compounds databases are based on the principle of "similar structural compounds present properties and similar activity" [10]. However the molecular form, functional groups, electronegativity, and a great number of properties distort this principle giving place to the appearance of outliers appearance and drops correlations in the models that are only based on similarity measures.

In this work an algorithm for the extraction of patterns in fingerprints is described. The algorithm, in a 4-stage process, extracts common subarrays to all or fingerprint subsets representing the group of chemical compounds of the dataset. The algorithm was completely developed in Java and builds an efficient hierarchical data structure to represent the group of patterns present in the fingerprints of the dataset and the relationships between them. This structure allows us to analyze the dataset in different abstraction levels associating molecules to each node of the hierarchy and, besides that the creation of efficient patterns matrix structures for the construction of classification models and the prediction of physical-chemical properties and biological activity.

The paper has been organized in the following way: after the introduction, in section 2 the developed algorithm is described and in section 3 the algorithm evaluation is show. Finally, a section of conclusions is included.

II. ALGORITHM DESCRIPTION

The objective of the algorithm is the extraction of patterns in a dataset of chemical compounds represented by its corresponding fingerprints. A pattern is a subarray of a fingerprint that it is common to one or several elements (fingerprints) of the dataset.

For the generation of the fingerprints corresponding to the dataset, the library for fingerprint generation CDK (Chemistry Development Kit) has been used [*11*].

In the generation of the fingerprint, three parameters can be adjusted: a) *Fingerprint Length*: The size of the vector of bits that represents each fingerprint, for a better characterization of the molecular structure a minimum value amount 512 to 1024 bits is recommended, b) *Search Depth*: It represents a numeric value that determines the refinement of the process for the relationships between molecular substructures and their corresponding representation in the fingerprint (a 512 value is recommended), c) *Path Length*: This parameter determines the highest value in the path considered in the molecular structure to construct the fingerprints.

The proposed algorithm extracts the patterns in a 4-stage process creating a hierarchical classification structure. In this hierarchical structure each node stores a pattern using a fingerprint format (array of bits) that represents the common bits "on" (1 value) between a fingerprint group and a list with the elements (molecules) of the dataset that satisfy this pattern. The different levels in the hierarchical structure store patterns obtained by the refinement of the nodes of the previous levels, reaching the leaf levels in the structure where the refinement is not possible. The process of pattern extraction is carried out without loss of information, so that the original dataset of fingerprints can be reconstructed traveling along the hierarchical structure.

Figure 1 shows the principal activities that compose the proposed algorithm.

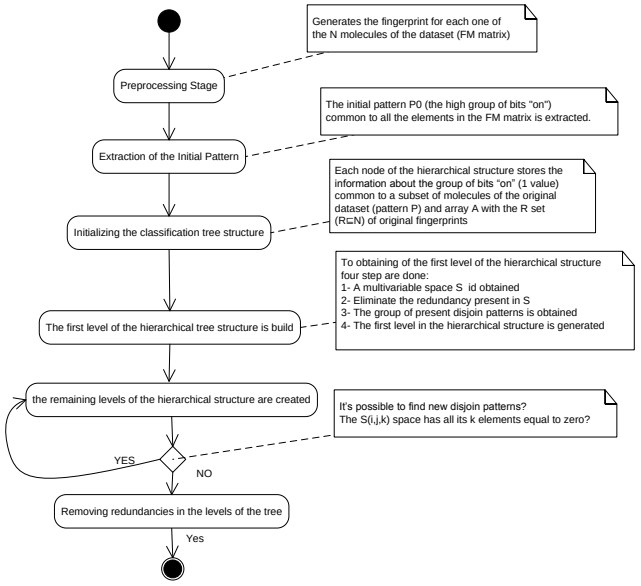

Fig. 1. Fingerprint-based classification algorithm.

The preprocessing stage generates the corresponding fingerprint for each one of the N molecules of the dataset. The result of this process is the matrix *(FM)*, with *N* rows, and *L* columns, *L* being the longitude of the fingerprints. Each element *(i, j)* of the *FM* matrix corresponds to a bit of the fingerprint of the molecule *i*, taking values 0 or 1.

The processing phase is carried out with the following steps:

1. ***Extraction of the Initial Pattern (P0)****:* The initial pattern *P0* is a vector which a size similar to L that has the bigger group of bits "on" (ones in the fingerprint) common to all the elements in the *FM* matrix. This information must be used to initialize the hierarchical data structure used in the classification process. An element of patron *P0 (k)* = 1, if all the elements of the matrix *FM(i, j)*=1, $\forall 1 \leq i \leq N, 1 \leq j \leq L, k = j$; in other case $P_0(k) = 0$.
   Due to the fact that all the elements of the *FM* matrix satisfy the pattern *P0*, the bits considered should be ignored for the following steps of the process. In order to do so, an initial matrix of classification (IMC) is built using the operation described in the equation (1):

$$IMC(i,j) = FM(i,j) \wedge [NOT(P_0(j))] \qquad (1)$$

2. ***Initializing the classification tree structure***: The hierarchical structure used by the classification algorithm is formed by nodes that represent disjoint bit patterns in the FM matrix; two patterns are considered disjoint if they differ in at least one of their bits. The P0 pattern extracted in the previous stage is used as node root in this structure. Each node of the hierarchical structure stores the information about the group of bits "on", common to a subset of molecules of the original dataset (pattern P) and array A with the R set ($R \sqsubseteq N$) of original fingerprints (dataset fingerprint) that satisfies the pattern assigned to the node. In the root node A_0 has the same elements as the IMC matrix
3. ***Building of the first level of the hierarchical tree structure***: In order to obtain the first level of the hierarchical structure, the following tasks are done using the A_0 array:

   a) A multivariable space S it is obtained carrying out the dot product of the fingerprints indexed by the A_0 array (root node). This space S can be represented as a symmetrical and three-dimensional matrix $S_{(i,j,k)}$, where: i and j represent each one of the fingerprint in the A_0 array, and k it represents the result of the

"AND" operation between the fingerprint i and j, which means that:

$$S = A_{ij} \wedge A_{ij} \; \forall \; i \neq j \quad (2)$$

b) The following step is the elimination of the redundancy present in S, which means the elimination of those identical patterns with the purpose of obtaining a disjoint patter representation space ($\overline{S}$).

c) Using $\overline{S}$, the group of present disjoint patterns is obtained. Each P pattern is formed by groups of "on" bits common (or not) to R disjoint subsets of molecules present in the A_0 array.

d) The first level in the hierarchical structure is generated. Each node of this level stores a pattern $P_{1,j}$ obtained in the previous step (j is the pattern's index in the resulting disjoint pattern group), and an array $A_{1,j}$ containing the references to the fingerprints of the $R_{1,j}$ subset of elements of the array A_0 that pattern $P_{1,j}$ satisfies.

The process of eliminating redundancy to obtain the $\overline{S}$ disjoint space considers that two elements of S are different if they differ at least in the value of one of the bits. At the same time that the redundancies are eliminated, the space is ordered so that the fingerprint (pattern) of the element $\bar{S}_{(i,j,*)}$ has a high cardinality (great number of bits "on") that the fingerprint (pattern) of the element $\bar{S}_{(i+1,j,*)}$.

In the $\overline{S}$ space elements may appear that, although not redundant, their structure is a subset of another element, that means: $\bar{S}_{(i,j,*)} \subset \bar{S}_{(p,q,*)}$. In this step it is necessary to eliminate these elements and, in order to carry it out, all the couples of $\bar{S}_{(i,*,*)}$ elements are compared, taking into account the fact that the comparisons between the $\overline{S}$ elements are symmetrical, and if $\bar{S}_{(i)} \wedge \bar{S}_{(i+1)} = \bar{S}_{(i+1)} \rightarrow \bar{S}_{(i+1)} \subset \bar{S}_{(i)}$, the pattern $\bar{S}_{(i)}$ *is* upgraded compared to the pattern $\bar{S}_{(i+1)}$ and continuing the process with the following pattern. Otherwise, you continue with the process with the following element.

This process is repeated for all the possible pairs, and at the end the $\overline{S}$ space contains a group of not redundant, ordered and disjoint elements that will be used for the construction of the first level of the tree.

Once the $\overline{S}$ space is generated, the different patterns are assigned to the nodes of the first level of the hierarchical structure. The orderly creation of $\overline{S}$ guarantees that pattern $P_{(1,i)}$ will have a great or the same cardinality (a great number of bits "on") than the pattern $P_{(1,i+1)}$, and so on.

Fig. 2. Pattern Classification tree

When a node of the hierarchical structure is created an array $A_{1,i}$ is assigned, storing the subsets of molecules which satisfy the pattern represented by the node, being a subset of the array associate to the parent node. Figure 2 shows an example of the structure of the classification tree.

To allocate molecules to the tree nodes two methods were tested:

- *Multiple allocations* (*MA*). A molecule can be assigned to more than one pattern or tree node of a given tree level. In the process the molecule fingerprint in the IMC matrix is updated with respect to the pattern of the node, and the bits "on" are deleted so that the cardinality of the fingerprint is reduced for the following stage of the algorithm.
- *Simple allocation* (*SA*). In this case, the molecule can only be assigned to just one node of the tree. This type of allocations improves the execution time of the algorithm, because it produces classification trees with lower number of nodes. In this method, the fingerprint of the molecule stored in the IMC matrix and assigned to the node is updated with respect to the pattern, reducing the fingerprint cardinality in the next stage of the algorithm.

4. ***Building of the remaining levels of the hierarchical structure***: The construction of the remaining levels of the tree is carried out according to stage 3. In this process, the pattern assigned to the node for which the following level has been generated is used as the initial pattern. The process is completed traveling the hierarchical structure, in

this way, until the $n+1$ level is totally generated the $n+2$ level does not begin.

5. ***Removing redundancies from the* tree *levels***: When multiple allocation method is used, once the first level of the tree is completed, the *A* arrays associated with the *P* patterns are analyzed with the aim of erasing redundancies at the same level and reducing the number of patterns.
   If two nodes in a new level with full disjoint patterns P_a y P_b have identical lists of associated molecules, that is, identical value of the associated arrays A_a=A_b, then both patterns are joined and reduced to just one pattern $P_c = P_a \vee P_b$, with an associated array $A_c = A_a = A_b$.
6. ***End of the processing stage***: The iterations to construct the hierarchical structure stop when it is not possible with to extract new patterns from the *A* array associated to a given node, that is, the *S(i,j,k)* space has all its *k* elements equal to zero.

Having finalized the building tree process, a set of fingerprint patterns distributed in a hierarchical structure will have been obtained. Each node of this tree contains a *P* pattern consisting of an array, with the same structure as the original fingerprints. The bits set "on" in a pattern corresponds to the bits set "on" in the subset of fingerprints referenced by the *A* array associated to the node.

All tree nodes contain disjoint patterns, although a molecule might be referenced for more than one *A* array associated to a pattern. This characteristic of the hierarchical structure allows us to consider the data set at different abstraction levels, depending on the tree level selected. Thus, it is possible to consider different complexity of patterns and therefore different classification levels of the data set.

Figure 3 shows the tree structure obtained for thirty seven molecules using a fingerprint size of 1024. Attached to each node is shown the pattern cardinality (number of bits "on") and the list of molecules associated to the pattern (the information stored in the *A* array).

The root node has a cardinality of 137 (the pattern consists of a fingerprint with 137 bits "on". In the first level, eight patterns have been extracted, with cardinality from 129 for N_{11} node to 7 for N_{16} node. These patterns are refined until the last level of the tree has been reached.

We observe in Fig. 3, for instance, molecules 36 and 37 have two common patterns: N_0 and N_{11} and, therefore, 137 + 97 = 234 common bits. Lastly these patterns are classified into distinct nodes, showing no more bits common are present in these molecules.

The inspection of the tree at different levels permits the building of different clusters of the original data set. These clusters can be refined by the consideration of lower tree levels. Thus, in the first level we observe a cluster defined by N_{12} pattern and composed by 19 molecules. The next level of the tree groups six new clusters (N_{121}, N_{122}, N_{123}, N_{124}, N_{125} y N_{126}) composed by 3, 3, 2, 1, 4 y 6 molecules respectively. In the next level, the N_{121} cluster is split into two new clusters (N_{1211} y N_{1212}). Thus, the refinement of the clusters in the building process of the tree is finalized when it is not possible to refine any more clusters at a given level.

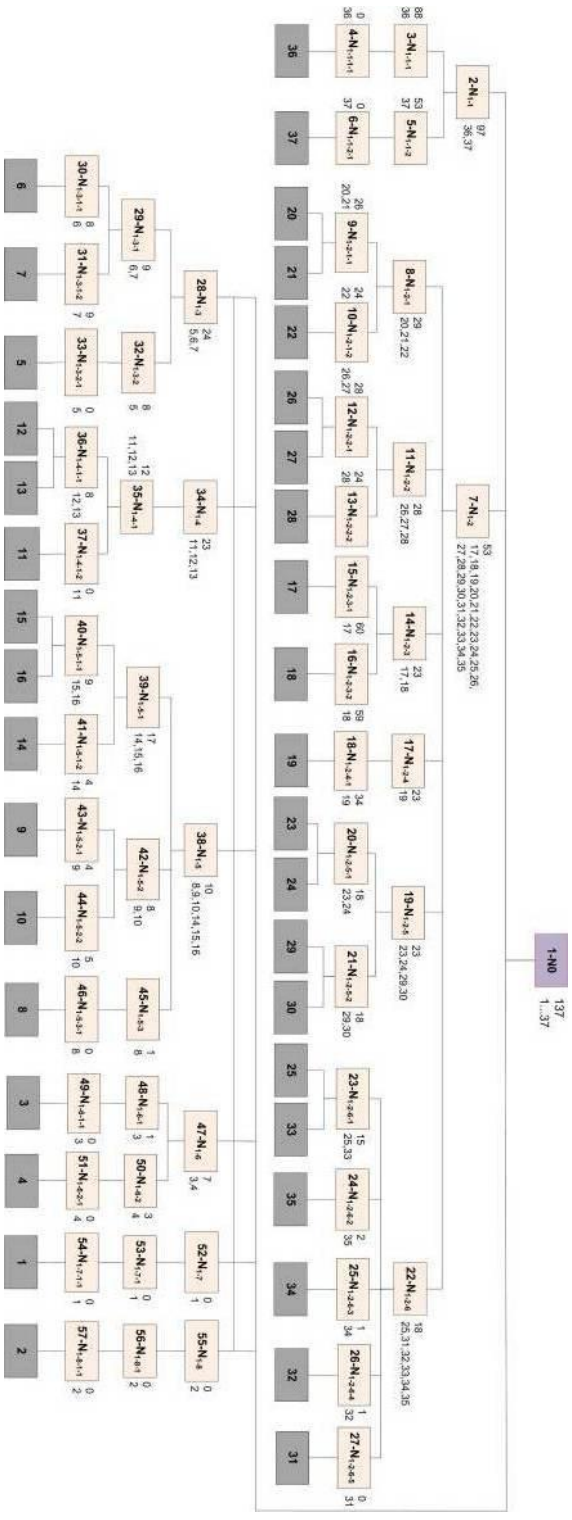

Fig. 3. Example of Classification tree

III. EVALUATION OF THE FINGERPRINT-BASED CLASSIFICATION METHOD

The algorithm has been completely developed in Java and integrate in CoChiSE software [*12*]. CoChiSE is software, oriented to computational chemistry applications and integrating utilities for database management, calculation of molecular descriptors, similarity measurements and QSPR/QSAR applications, among others. Table 1 summarizes some algorithm parameters tested.

For all the considered data sets, the processing time necessary for building the tree is directly proportional to the number of molecules, as well as the fingerprint length. For some tests, the processing time for fingerprint of size 1024 is somewhat lower than for fingerprint of 512 bits. This fact in some cases shows that fingerprint lengths of 512 are enough to represent all the information, and higher lengths do not provide more information.

The simple allocation method improves the execution time and reduces the number of nodes of the classification tree. This behavior can be observed for fingerprint lengths of 512 and 1024 in data sets as: a) *Quinones*: a reduction of the 1.63% using simple allocation method, d) Enaminones: a diminishing of 14.82 % with simple allocation.

TABLE I. Experimental results for tree building and runtime in milliseconds: (1) breadth-first method (multiple allocation method), (2) breadth-first method (simple allocation method), $|P_0|$: cardinality of the initial pattern, F size: size of the fingerprint, Path size: minimum-maximum path length

Dataset	Elements	Path size	F Size	$\|P_0\|$	Nodes		Levels		Runtime	
					(1)	(2)	(1)	(2)	(1)	(2)
Cyclopentenes	271	22-32	512	34	326	253	6	7	1829	1687
			1024	24	297	236	8	7	6719	6016
Quinones	74	18-34	512	127	80	63	5	5	157	156
			1024	100	79	57	6	4	562	578
Enaminones	37	15-28	512	120	219	33	7	5	141	141
			1024	137	143	32	7	5	125	62

An important issue is comparing the similarity between molecules assigned to a same cluster. In order to evaluate this characteristic we calculated the average MCS [*13*] similarity of each cluster considering the molecules assigned, and the new value considering when every other molecule of the dataset might be assigned. Thus, the generated clusters have assigned the molecules correctly; they will have a higher similarity than if other molecules had been assigned.

To evaluate the quality of a clustering result, an index is introduced to assess whether molecules interacting with the same activity lie in the same subtree. We calculated an enrichment factor (*EF*) for each cluster [*14*], which gives an estimate of how well compounds that bind to the same target (or class) are clustered in a dendrogram node *i* (equation 3)

$$EF_{(i,c)} = \frac{N_{(i,c)}}{N_c} \Big/ \frac{N_i}{N} \quad (3)$$

with $N_{(i,c)}$ being the number of entries in node i belonging to class c, N_i being the total number of entries in node i, N_c being the total number of entries of class c in the data set, and N being the overall number of entries. EF > 1 indicates that more compounds belonging to the activity class c are clustered in a tree node than expected from an equal distribution.

To obtain a generalized view on the distribution of molecules interacting with the same receptor target in the overall cluster dendrogram, we suggest the following: For a dendrogram level we calculate the average of EF (Av_{EF}) of all n (n is the number of clusters) EFs of class c, which are larger or equal to one (equation 4).

$$Av_{EF} = \frac{1}{n}\sum_i EF_{i,c}; EF \geq 1 \quad (4)$$

Average enrichment factors are calculated for all levels, where the number of clusters is less or equal to the number of molecules belonging to c class.

Of the three studied datasets the Quinones family was previously studied due to its antifungal activity in humans [15]. The biological activity was measured in terms of their MIC50 values. Two classes well defined of compounds (active and not active) were considered in [15]. The active set is composed of 54 elements (class 1) and the remaining 20 are low or non-active compounds (class 2).

Considering the average enrichment factors for quinones of class 1, the proposed algorithm was compared to classical clustering methods hierarchical Ward and kmeans [16]. In this analysis the Av_{EF} values was calculated using two different n values corresponding to the number of nodes for the two first levels in the pattern hierarchic. For the first level the propose algorithm ($Av_{EF} = 2.1$) reports a better behavior than ward ($Av_{EF} = 1.8$) and kmeans ($Av_{EF} = 1.2$). The values of Av_{EF} for the second hierarchical level increase to the first level, while also showing a better behavior of the proposed algorithm compare to the classical algorithms analyzed (2.7 for the proposed method, 2.3 for Ward and 1.7 for kmeans).

IV. CONCLUSIONS

The molecular activity does not only depend on the structural similarity of compound, the presence/absence of functional groups, and other properties related to these groups are closely related to the molecular activity. The propose algorithm in this work based on fingerprint patterns extraction is a good approach for the chemical database clustering, showing a better behavior than some classical methods. The use of simple allocation method improves the execution time and it

reduces the number of nodes of the classification tree. Considering the hierarchical representation it is possible to create a new representation space composed by pairs <patterns, molecule> without redundancy. Using this structure, different matrix representations can be generated, and these new representation spaces are useful for studying and predicting the structure/activity relationship (QSPR / QSAR) of molecules. We will therefore focus our future works in this direction.

REFERENCES

[1] J. W. Raymond, et al., "Comparison of chemical clustering methods using graph- and fingerprint-based similarity measures," Journal of Molecular Graphics & Modelling, vol. 21, pp. 421-433, 2003.

[2] S. C. Basak, et al., "A graph-theoretic approach to predicting molecular properties," Mathematical and Computer Modelling, vol. 14, pp. 511-516, 1990.

[3] P. Guttiérrez Toscano and F. H. C. Marriott, "Unsupervised Classification of Chemical Compounds," Journal of the Royal Statistical Society: Series C (Applied Statistics), vol. 48, pp. 153-163, 1999.

[4] M. Vogt and J. Bajorath, "Bayesian Screening for Active Compounds in High-dimensional Chemical Spaces Combining Property Descriptors and Molecular Fingerprints," Chemical Biology & Drug Design, vol. 71, pp. 8-14, 2008.

[5] A. Kumar and M. I. Siddiqi, "Virtual screening against Mycobacterium tuberculosis dihydrofolate reductase: Suggested workflow for compound prioritization using structure interaction fingerprints," Journal of Molecular Graphics and Modelling, vol. 27, pp. 476-488, 2008.

[6] B. Ivan P, "Generation of molecular graphs based on flexible utilization of the available structural information," Discrete Applied Mathematics, vol. 67, pp. 27-49, 1996.

[7] R. Raveaux, et al., "A graph matching method and a graph matching distance based on subgraph assignments," Pattern Recognition Letters, vol. 31, pp. 394-406, 2010.

[8] M. Gary L, "Graph isomorphism, general remarks," Journal of Computer and System Sciences, vol. 18, pp. 128-142, 1979.

[9] M. Dehmer and F. Emmert-Streib, "Comparing large graphs efficiently by margins of feature vectors," Applied Mathematics and Computation, vol. 188, pp. 1699-1710, 2007.

[10] G. M. Maggiora, et al., "Looking for buried treasures: The search for new drug leads in large chemical databases," Mathematical and Computer Modelling, vol. 11, pp. 626-629, 1988.

[11] JChem, ed: version 5.3.7. Chemaxon Ltd, 2010.

[12] B. Palacios-Bejarano, et al., "An Open Environment to Support the Development of Computational Chemistry Solutions im AIP Conference Proceedings," in AIP Conference Proceedings, 2009, pp. 519-522.

[13] G. Cerruela-García, et al., "Step-by-step calculation of all maximum common substructures through a constraint satisfaction based algorithm," Journal of Chemical Information and Computer Sciences, vol. 44, pp. 30-41, 2004.

[14] R. O. Duda, et al., Pattern Classification: John Wiley & Sons, 2000.

[15] S. Y. Choi, et al., "The development of 3D-QSAR study and recursive partitioning of heterocyclic quinone derivatives with antifungal activity," Bioorganic & Medicinal Chemistry, vol. 14, pp. 1608-1617, 2006.

[16] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, pp. 651-666, 2010.