Hindawi Publishing Corporation
Journal of Automated Methods and Management in Chemistry
Volume 2009, Article ID 502527, 12 pages
doi:10.1155/2009/502527

Research Article

Interactive Data Mining for Molecular Graphs

Burcu Yılmaz[1,2] and Mehmet Göktürk[1]

[1] *Department of Computer Engineering, Gebze Institute of Technology, 101 41400 Kocaeli, Turkey*
[2] *Department of Computer Engineering, Istanbul Kültür University, 34191 İstanbul, Turkey*

Correspondence should be addressed to Burcu Yılmaz, byilmaz@gyte.edu.tr

Received 1 September 2009; Accepted 2 October 2009

Recommended by Peter Stockwell

Designing new medical drugs for a specific disease requires extensive analysis of many molecules that have an activity for the disease. The main goal of these extensive analyses is to discover substructures (fragments) that account for the activity of these molecules. Once they are discovered, these fragments are used to understand the structure of new drugs and design new medicines for the disease. In this paper, we propose an interactive approach for visual molecule mining to discover fragments of molecules that are responsible for the desired activity with respect to a specific disease. Our approach visualizes molecular data in a form that can be interpreted by a human expert. Using a pipelining structure, it enables experts to contribute to the solution with their expertise at different levels. In order to derive desired fragments, it combines histogram-based filtering and clustering methods in a novel way. This combination enables a flexible determination of frequent fragments that repeat in molecules exactly or with some variations.

1. Introduction

Design of new medical drugs is an exhaustive process with many challenges. Molecular fragment mining is one of the vital stages of the process. Researchers design new drugs using features extracted by molecular fragment mining methods. These methods are used to discover relationships between structure and activity of the compounds. That is, they are used to examine different compounds with a desired activity and extract some common substructures that provide this activity. The common methodology is called Structure Activity Relationships (SAR).

Several methods are proposed to determine a mathematical equation correlating different properties of the compounds. These methods are called as Quantitative Structure Activity Relationships methods. Several other methods known as Qualitative-SAR methods examine different compounds and extract some common significant substructures with desired activity. In this paper, the main focus is a Qualitative-SAR method that uses active fragments as a map to guide for molecular design [1]. An active fragment (pharmacophore) is a set of similar structural features in the structure of active molecules (or drugs) that is responsible for their biological activity.

Researchers usually model and visualize compounds using 3D graph representations. A *graph* is a kind of data structure that consists of a set of nodes and a set of edges between them. Graphs are used to represent objects whose individual elements are interconnected in complex ways. For example, in a 3D fully weighted molecular graph representation, each node may represent characteristic features of atoms (e.g., electrical properties), and each edge may represent characteristic features of bonds between them (e.g., length, value of Wiberg's index, and so on). Therefore, researchers study frequent subgraph mining approaches [2, 3] to find frequency of common fragments (e.g., pharmacophores), which is a collection of bonds and atoms in the structure of compounds with the same activity for a specific disease. The success of the proposed methods relies on the features used for the representation. As the complexity of representation increases, interpretation of the represented molecular information becomes harder, because the graph-based approaches usually have high complexities. For example, while comparing two molecules in terms of their structure and activity, it may be necessary to compare their substructures, which are represented as graphs. Hence the problem of comparing two molecules turns into comparing two graphs. Unfortunately, deciding whether two

graphs have identical topological structures or not has an unknown complexity [4]. The subproblem that is deciding whether one graph is a subgraph of another or not is known to be NP-complete [4]. Therefore, frequent subgraph mining approaches may not find exact solutions in a bounded time because of their high complexities. Researchers use heuristics to reduce search space while solving subgraph mining problems [5, 6]. In this way, they try to get rid of high computational burden inherent to these problems. For example, SUBDUE algorithm uses a greedy search technique for frequent subgraph mining [5] to reduce the time complexity with a possible cost of missing some important substructures.

As stated above, many researches convert the problem of discovering frequent substructures in active molecules into a frequent subgraph discovery problem, in order to use graph mining approaches. However, current graph mining approaches can determine frequent subgraphs only if these subgraphs repeat exactly the same in a graph database. Therefore, in many settings, these methods may not find the most informative substructures that do not exactly repeat in the molecules but exist in those molecules with some fine differences. Furthermore, because these approaches work like black boxes, domain experts cannot easily interrupt the process to incorporate their knowledge into the solution. That is, another problem related to current substructure discovery approaches is their blind calculations. Therefore, even if these approaches find solutions, their solutions should be extensively examined, tested, and verified at the end by the domain experts. This leads to a system where any failure in the early stages of the process cannot be corrected until the overall process is completed. Alternatively, in this paper, we advocate directly incorporating domain knowledge into the solution process at different levels. For this purpose, we propose to use a data pipelining approach, where domain experts can intervene the overall process to enhance the solution with their knowledge and expertise.

More specifically, in this paper, we propose a visual data mining method for frequent substructure extraction, where histogram-based filtering method is used to reduce the search space. In our approach, graph representations of molecules are projected into a 3D feature space (named *atom-bond-atom space*). Each bond of a molecule (an edge with nodes at each end in the graph) is represented as a point in this space. When we project all of the molecules with a certain activity into this space, the resulting points compose clusters of bonds that are repeated in the structure of the active molecules (possibly with fine differences). However, discovering these clusters is nontrivial because of the noisy points that represent infrequent bonds. At this stage, we filter the noise using a histogram-based visual data mining method so that the clusters can be discovered more clearly by various clustering algorithms. Once the clusters are discovered, frequent substructures of active molecules are computed. During all these steps, a domain expert can intervene to guide the system if necessary (e.g., during filtering, clustering, and so on). We demonstrate how our approach can determine frequent substructures of molecules step by step through a case study, using the tuberculosis dataset from literature [7, 8]. We empirically compare our approach with other approaches from literature. Our experiments show that our approach can successfully determine active fragments that account for the activity of those molecules. We lastly show how the determined fragments can be used as features to automatically categorize new molecules as active or inactive with respect to a specific disease.

The rest of this paper is organized as follows. In Section 2, we overview related work. In Section 3, we describe the proposed approach in detail with examples. In Section 4, we experimentally evaluate the performance of the proposed approach with respect to other approaches from literature. Lastly, we conclude our paper in Section 5.

2. Related Work

There are two groups of SAR methods. The first one is known as quantitative-SAR, which derives a correlation between the descriptors of molecules and their activity. Molecular description vectors are prepared for every molecule. Then, different machine learning techniques are applied to the prepared dataset as described in literature to learn how to classify molecules. Linear discriminant analysis (LDA), multilayer perceptrons, support vector machines (SVM), k-nearest neighbors (k-NN), simulated annealing, partial least squares, linear regression, and classification trees are the most studied methods on quantitative-SAR for classification of molecules [9, 10].

Qualitative-SAR is the second group of approaches which is interested in finding some common substructures in the structure of molecules. There are many approaches related to qualitative-SAR in literature which are based on frequent substructure mining methods. These methods will be mentioned in this section with sufficient detail. Some qualitative-SAR approaches use graph mining methods to find common substructures that exist in active molecules with high probability, but exist in inactive molecules with low probability [2, 3].

Our research is related mainly to two important research areas: frequent substructure extraction and visual data mining. Especially in qualitative-SAR applications, graph-based data mining approaches are used to find frequent substructures (subgraphs) in graph-based mass datasets. These methods struggle with two important problems: graph isomorphism and subgraph isomorphism [11]. Graph isomorphism is the problem of deciding whether two graphs have identical structures (e.g., finding common fragments from two molecules). It has an unknown computational complexity [4]. The second problem, subgraph isomorphism, is the problem of deciding whether one graph is a subgraph of another graph. This problem is known to be NP-complete [4]. Graph-based frequent substructure extraction methods avoid this high complexity using some limitations.

In recent years, a number of algorithms have been developed for frequent substructure mining. They rely their approaches on candidate subgraph extraction processes and pruning some of them using some methods or representations. SUBDUE is one of the well-known graph-based frequent substructure mining approaches. It uses greedy

search to avoid high complexity of graph isomorphism. Simplifying the data by compressing repetitive substructures, SUBDUE method finds frequent substructures within the data. It also enables abstraction of detailed and complex structural data by iteratively constructing a hierarchical description of it. The algorithm first searches a single vertex that matches with the substructure. At each iteration, algorithm expands matching substructure using a greedy selection of best suitable neighbor edges until all possible substructures are found. SUBDUE finds an incomplete set of frequent substructures, because of the greedy idea behind it.

Some other approaches depend on the principle of a priori algorithm in basket analysis [12]. A priori-based frequent substructure mining methods extract association rules which have higher support and confidence than those of a threshold value. Various techniques are used to reduce the subgraph isomorphism computations. Canonical labeling representations of graphs are mostly used for this purpose. For example, to minimize computation and storage, frequent subgraph discovery (FSG) algorithm also uses canonical representations [6]. Hence, this method is suitable for small and sparse graphs. A graph can be represented with many different canonical labels depending on the orders of vertices and edges. Therefore, FSG searches all permutations of vertices to find a unique canonical label. To narrow down the search space, vertex invariants technique that partitions the graph into subgraphs (not mentioning candidate frequent subgraphs) is used. Then all permutations of vertices are calculated inside the graph partitions. FSG generates candidate subgraphs and increases the size of the candidate subgraphs by adding one edge to each candidate at each time. Using a breadth-first approach, it discovers the lattice of frequent subgraphs. Frequencies of achieved candidate subgraphs are calculated to prune the subgraphs that do not satisfy the support constraint. For each candidate, a mapping with the other graph is searched to solve the graph isomorphism problem using canonical representations.

Graph-based substructure pattern mining (gSpan) transforms the problem into finding common related parts in Depth-First Search (DFS) codes to find frequent subgraphs. It uses depth-first search tree structure and DFS lexicographic order to find frequent subgraphs. In this approach, each graph has a DFS code, and minimum DFS codes are used for comparison of two graphs. *gSpan* provides efficient computational time and memory consumption [13]. MoFa uses a priori method in basis. It uses a tree structure where each node represents a subgraph. There is a counter in each node that shows the number of graphs including the subgraph. To prune the search space in the tree structure, two pruning methods are used. In the first pruning method, it removes nodes which have a counter value smaller than support threshold value. In the second one, it removes nodes which have a number of nodes more than the desired number of nodes. For every graph, Gaston uses a "graph code" that shows the order of nodes and edges joined. It also uses a tree structure where each node represents a subgraph and uses depth-first search in the tree structure. The methodology prunes the tree structure by using a priori property [12]. Inductive logic programming (ILP) is one of the well-known techniques in graph mining [14]. ILP derives logic-based rules to identify frequent substructures. Although many graph-based searching approaches focus only on instances from only one class, ILP uses observations from every class. However, it generally does not support numerical calculations and cannot model quantitative classification problems.

3. Interactive Mining of Molecules

In this section, we propose a novel approach to discover frequent fragments of active and inactive molecules. Previous researches mostly use graph-based frequent substructure mining methods to discover frequent active fragments that account for the activity of the molecules for a specific disease [3, 11]. However, these methods work like black boxes where the data with some parameters are fed in the beginning and the solution is outputted at the end. Hence, domain experts cannot easily interrupt the process to incorporate their knowledge into the solution. This leads to two main problems: (1) the found solutions should be extensively examined by the domain experts at the end and (2) any failure in the early stages of the process cannot be corrected by the domain experts until the overall process is completed. In this paper, we advocate directly incorporating domain knowledge into the solution process at different levels. For this purpose, we propose to use a data pipelining approach, where domain experts or users can intervene the overall process to enhance the solution with their knowledge and expertise.

3.1. Overview. Figure 1 shows the data flow diagram of the proposed data pipelining structure, where raw molecular data from a dataset are filtered iteratively. The dataset contains 3D graph representations of molecules that are either active or inactive with respect to a specific disease, for which a new drug is to be designed. With the proposed data pipelining, a domain expert or a user is able to provide critical information to the system at the intermediate steps. At the beginning, the user gives some initial parameters using her/his expertise. Using these parameters and the dataset, the system computes two histogram-based visual representations as described in Section 3.2: activity map and inactivity map. Before the fragment mining begins, activity and inactivity maps are displayed to the user to show information about candidate frequent fragments for the active and inactive molecules, respectively. If the candidate frequent fragments are not clearly seen from the visual representations, the user can change the parameters. Once the resulting maps are satisfactory for the user, infrequent and insignificant substructures in the data are filtered automatically using the proposed approach in Section 3.2. After filtering process, residues give the most significant candidate frequent fragments. Then, the proposed method transforms filtered data to atom-bond-atom space, where each point represents features of a bond. The resulting points in this space constitute clusters that correspond to frequent fragments. These clusters can be determined

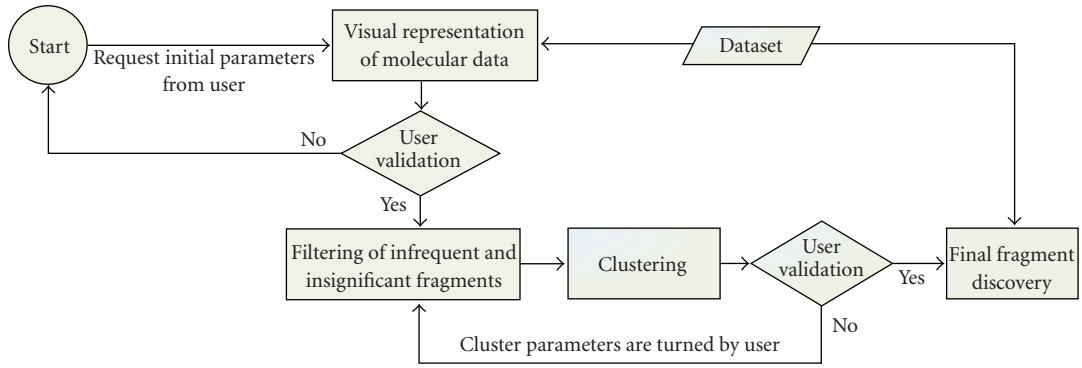

Figure 1: Data flow diagram.

by a suitable clustering algorithm easily as described in Section 3.4, because the filtering step eliminates insignificant and infrequent fragments that constitute the noise between the clusters in atom-bond-atom space.

At the initial step of clustering, clustering parameters are calculated automatically by the system. Then, clusters are determined using these parameters. The found clusters are converted to the corresponding 3D fragments and visualized to the user. Using the advantage of data pipelining structure, the user can change the automatically calculated parameters of the clustering algorithm depending on her/his expertise after analyzing the visualized 3D fragments. As a result, she/he either confirms these clusters or tunes the clustering parameters to force the system to recalculate clusters depending on the new parameter values. Once the user confirms the computed clusters, unfiltered data and cluster information are used for fragment discovery as described in Section 3.4. Because we use the whole dataset (unfiltered data) together with the cluster information, fragment discovery is not significantly affected by the data loss caused by the filtering process.

Using the proposed pipelining structure, we enable users to enhance the solution with their domain knowledge and expertise. If the users are not confident with their expertise, they can simply confirm the automatically calculated intermediate results (e.g., discovered clusters). In this case, the overall system works almost like a fully automated process.

3.2. Visualization of Molecule Properties. Graphs are one of the most frequently used representations in molecule visualization [3]. However, this visualization technique can indicate only information about topology of molecules. Although edges and nodes are used to represent some properties of molecules, it is inefficient to compare different properties of molecules using 3D graph visualization.

In this paper, a transformation from 3D graphs to molecular information visualization is implemented. To make this representation more understandable to the experts, images that display information about molecules are created. These images contain topological information of molecules as well as additionally requested properties. To represent molecular graphs, *electron-topological matrices of conjugency (ETMC)* are used [15]. In this method, a molecule with n atoms is represented with a number of upper triangular fully weighted ETMC matrices, each representing various characteristics of molecules. A formulation of ETMC matrices is shown in (1). The main advantages of ETMC matrices are that their molecular representation reflect a molecule's electronic and 3D conformational properties and do not depend on the numbers and types of atoms:

$$\text{ETMC} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ & a_{2,2} & \cdots & a_{2,n-1} & a_{2,n} \\ & & \cdots & \cdots & \cdots \\ & & & a_{n-1,n-1} & a_{n-1,n} \\ & & & & a_{n,n} \end{bmatrix}. \quad (1)$$

In our study, diagonal elements $a_{i,i}$ $(i = j)$ of an ETMC matrix contain information about electronic properties of atoms, and nondiagonal elements $a_{i,j}$ $(i \neq j)$ include information about chemical properties of bonds between the corresponding atoms in ETMC matrix representation. If there is no bond, the distance between two atoms is used instead. For various characteristics of molecules (e.g., bond properties such as value of Wiberg's index), additional ETMC matrices are formed similarly in this manner. Hence, different descriptors of molecules can be examined for their accountability on activity for a specific disease.

For all integer values of i and j $(1 \leq i, j \leq n)$, vectors $[a_{i,j}, \min(a_{i,i}, a_{j,j}), \max(a_{i,i}, a_{j,j})]$ are obtained from ETMC matrices. Hence, corresponding molecules are split into pieces (bonds), where each bond is represented with a vector including information about the bond $(a_{i,j})$ and two atoms at each end ($a_{i,i}$ and $a_{j,j}$). These pieces are plotted with point pairs, $(a_{i,j}, \min(a_{i,i}, a_{j,j}))$ and $(a_{i,j}, \max(a_{i,i}, a_{j,j}))$, using a transformation from 3D to 2D coordinate system as shown in Figure 2. Thus, points are derived from an ETMC matrix and then projected onto 2D Cartesian system. In the following section, we propose a visual data mining approach that uses the points derived from ETMC matrices.

3.3. Gray-Scale Shading. In this section, we propose a visual data mining method called gray-scale shading. This method displays desired properties of molecules for the experts using 2D images. Therefore, we enhance the understandability of

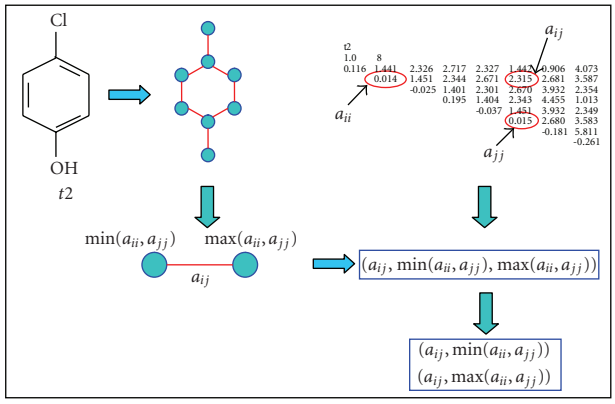

Figure 2: Transformation of data to 2D space.

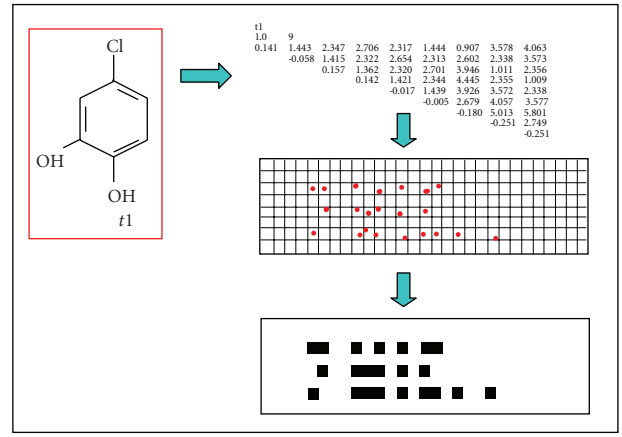

Figure 3: Resulting 2D image after processing an ETMC matrix.

the molecular information by the experts and enable the experts to compare different molecules easily in terms of their desired descriptors.

One of the problems while comparing molecules is that molecules posseses different degrees of flexibility. Because of this flexibility, even the topological structure of the same molecule can vary under different conditions. Therefore, values from ETMC matrices cannot be compared in a straightforward manner. To overcome this difficulty, we decrease the resolution of the data as follows. First, a 2D image is generated from the ETMC matrix of a molecule by dividing X, Y coordinate system into regular intervals in order to compose a 2D mesh. For this purpose, two parameters are used: Δ_x and Δ_y. Using the parameter Δ_x, X-axis of the coordinate system is divided into the intervals so that the width of each interval is Δ_x. Similarly, this is repeated for the Y-axis. The crossing intervals on the coordinate system compose a 2D mesh (a regular grid). In literature, irregular grids are also used as mentioned in Section 2. Usage of irregular grids results in various grids suitable for each molecule. But comparison of each molecule using different grids is difficult. Moreover, when finding similar frequent substructures, a general grid structure common for every molecule is needed. However, it is difficult to fix an irregular grid suitable for every molecule, because of the noisy data. Hence, not to increase complexity of process, a regular grid structure is used in this work. The resulting mesh is used to create a visualization of the molecule. Each point pair drawn from the ETMC matrix is plotted into the 2D mesh. Then, the rectangular regions of the mesh where the points fall are shaded. Figure 3 shows an example for the creation of a 2D image from an ETMC matrix.

In our dataset, we have two different disjoint classes of molecules: active molecules and inactive molecules. Using the methods described above, we produce images for each molecule in the dataset. Now, using these images, we describe how to compose activity and inactivity maps that show characteristic properties of active and inactive molecules, respectively. That is, each map shows the percentage of bonds that are common for the corresponding class of molecules. Hence, the frequent bonds in each class of molecules are seen easily on the related map.

For active molecules, an activity map is created using the aforementioned 2D mesh. However, this time, shades of the rectangles in the activity map denote percentage of molecules falling into the rectangular cells. For this purpose, we use a 2D histogram-based visualization method that shows frequent active fragments in the molecular dataset as follows. A rectangle $R_{I,J}$ can be considered in the 2D mesh, where $0 < I \leq N$ and $0 < J \leq M$. In order to determine the shade of each $R_{I,J}$ in the activity map, we average the shade of the corresponding rectangles $R_{I,J}$ in the images of the active molecules. In order to average shades, 8-bits gray scale is used where 0 corresponds to black. The resulting average value is converted to a gray scale. Hence, an activity map gives information about the parts of molecules that appear frequently in the structure of active molecules in a better perceivable format. The same methodology is used for creation of the inactivity maps, but this time, images from the inactive molecules are used, instead of images from the active molecules.

Resulting activity and inactivity maps provide important perceivable information to the experts using a histogram-based data visualization. For example, the activity map in Figure 4 shows the rectangles with darker and lighter shades. The dark fragments are the structures that are common to the most of the active molecules, whereas the lighter fragments represent the structures that are not repeating in the active molecules, so those fragments may not be significant for the activity. For various characteristics of molecules, various activity and inactivity maps can be formed. We filter the substructures of active molecules if these substructures fall into the lighter rectangles in the activity map. For this purpose, we use a threshold value determined by the expert. If the dataset is noisy (i.e, it highly contains infrequent fragments), a small threshold value will be chosen; otherwise, the chosen threshold value will be high.

In summary, first an activity map is constructed using all of the active molecules in the dataset, then algorithm determines the insignificant parts of each molecule in the

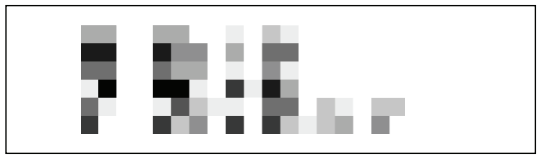

Figure 4: The activity map of 10 active molecules.

dataset using the decisions of expert (e.g., provided threshold). Lastly, the determined parts are filtered to eliminate redundant and noisy information. Alternatively, in respect to her/his domain knowledge and expertise, data falling into some regions on the activity map can be removed directly by the expert. The same procedure is applied to the computed inactivity map. Selection of Δ_x and Δ_y parameters does not affect the extracted active fragments significantly, because the filtered data give only preliminary information about activity and inactivity clusters. Nevertheless, several values of the Δ_x and Δ_y parameters should be tried for the best filtering and the best results.

3.4. Extracting Fragments Using Clustering. After filtering the data using activity and inactivity maps as described in Section 3.3, redundant and noisy bonds are removed from the active molecules in the dataset. The remaining bonds of the active molecules may contain the ones that compose active fragments, so we mine these bonds to discover these fragments. We determine active fragments of the active molecules as follows. First, the remaining bonds of each active molecule are transformed into 3D atom-bond-atom space, where features of each bond (and the atoms at its each end) are represented as a single point. Hence, an active molecule with m remaining bonds after filtering is represented with m points. Each point is in the form of $a_{i,j}, \min(a_{i,i}, a_{j,j}), \max(a_{i,i}, a_{j,j})$, where $a_{i,j}$, $a_{i,i}$, and $a_{j,j}$ are values from the ETMC matrix of the molecule and correspond to feature values of a bond and feature values of the atoms connected by this bond, respectively.

In this way, bonds of the active molecules are transformed into the 3D atom-bond-atom space. The result is a set of points that are distributed over the space, where the points representing bonds with similar features are in proximity. That is, the points are not distributed randomly in the space, but in a way that groups of points representing bonds with similar properties appear. We name these groups of points as candidate activity clusters, which are used to derive candidates of active fragments. In order to find candidates of activity clusters, we use average-link clustering method [16]. In this clustering method, initially each point in the space is regarded as an individual cluster. Then, clusters are merged iteratively according to the distances between the cluster centers. That is, two clusters are merged if their distance is smaller than that in a predefined threshold. Clustering is ended when there are not any two clusters that can be merged.

Each of found clusters is composed of points that represent bond patterns of active molecules. Hence, the determined clusters represent the substructures that exist in

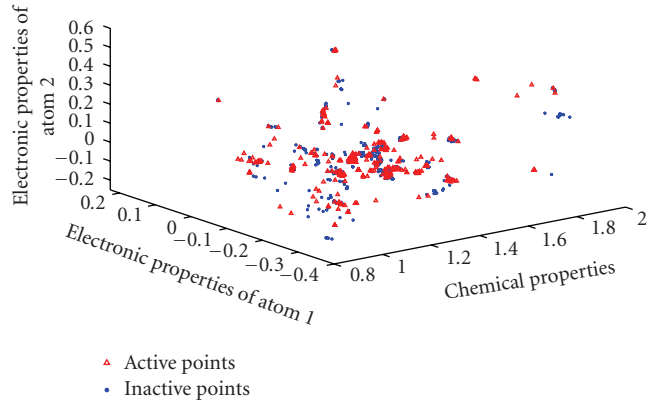

Figure 5: Unfiltered active molecules plotted on atom-bond-atom coordinate system.

active molecules. However, all of these substructures may not be considered as active fragments, because some of them may also be repeated in the inactive molecules. Active fragments should be the substructures that exist frequently in active molecules, but rarely in inactive molecules. This means that we are looking for clusters that are composed of bond patterns that are not repeated in the inactive molecules. Therefore, after determining clusters, for each cluster, we compute the percentage of active and inactive molecules that contain the molecular pieces in the cluster. The candidate clusters including higher percentages of active molecule bonds and small percentage of inactive molecule bonds are regarded as activity clusters. Bonds falling into active clusters are expected to compose active fragments.

In order to show advantage of the proposed filtering method, we show the points extracted from raw molecular data in Figure 5 and the filtered molecular data in Figure 6 in the 3D atom-bond-atom space. Although, Figure 5 gives a messy view of the molecular data, Figure 6 gives a cleaner view of clusters on the filtered data, because noise and uninformative points that show infrequent substructures (bonds) are removed from the data after filtering. In Figure 9, we show the resulting clusters on the filtered data with an example of molecule's two pieces falling into two activity clusters.

3.5. Computational Complexity. As we mentioned before, time complexity of subgraph isomorphism is NP-complete. Hence, heuristics are used instead of exact algorithms to solve frequent substructure discovery problem in graphs. In this paper, we also propose a heuristic that depends on interactive visual mining of molecules. This approach is composed of four sequential steps: activity map creation, filtering, clustering, and fragment extraction. Computational complexities of these steps are listed in Table 1. The highest complexity belongs to clustering step, where average-link clustering is used [16]. Complexity of the used clustering algorithm is $O(n^3 \times m^6)$, where n is the number of molecules in the dataset and m is the maximum number of atoms in a molecule. Therefore, the overall computational complexity of the proposed approach is only $O(n^3 \times m^6)$.

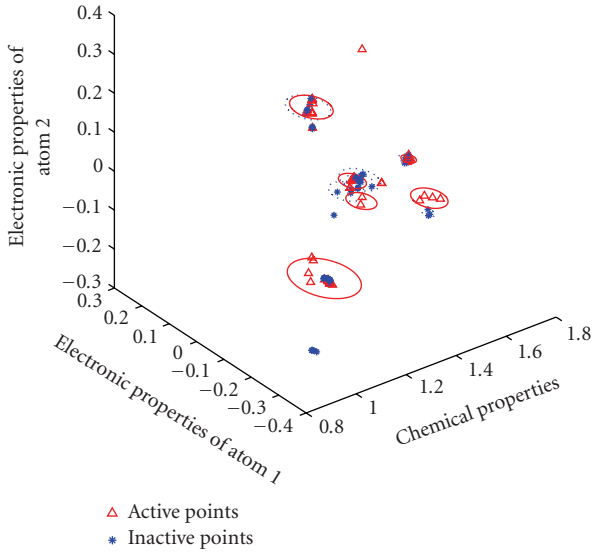

Figure 6: Activity and inactivity clusters extracted from filtered data.

Table 1: Complexity of each step in the proposed approach (n = number of molecules, and m = maximum number of atoms in a molecule).

Activity map creation	$O(n \times m^2)$
Filtering	$O(n \times m^2)$
Clustering	$O(n^3 \times m^6)$
Fragment extraction	$O(n \times m^4)$
Overall	$O(n^3 \times m^6)$

4. Evaluation

In order to demonstrate our approach better, we design realistic experiments with real-life data and simulations on the synthesized graphs. In our experiments, we have used antituberculosis dataset [7, 8]. This dataset is composed of 33 molecules (13 active and 20 inactive molecules). In order to examine our approach extensively, we also test it with synthetic datasets of varying sizes.

We demonstrate the performance of our approach in three steps. First, in Section 4.1, we show a case study to demonstrate how an expert can use our approach for creating activity and inactivity maps and deriving active fragments interactively. Second, using antituberculosis dataset and synthetic datasets, we empirically compare our approach with the well-known approaches from literature: SUBDUE [5], FSG [6], gSpan [13], Gaston [17], MoFa [3], and FFSM [18]. We have repeated each experiment 10 times and our results are significant according to 2-tailed t-test with 95% confidence level. We present our results in Sections 4.2 and 4.3. Third, in Section 4.4, we derive active and inactive fragments from the antituberculosis dataset. Then, we use the derived fragments as features and evaluate the performance of well-known classification methods in classifying molecules as active or inactive. Intuitively, if our approach is successful in determining active and inactive

Figure 7: Activity map.

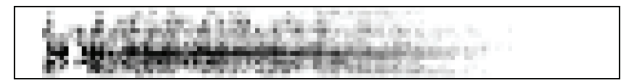

Figure 8: Inactivity map.

fragments that account for the activity and inactivity of the molecules, then the classification methods using those fragments as features are expected to demonstrate a good performance.

All the experiments of the proposed method are tested on 1.6 GHz Intel Pentium Core II Duo, 1 GB Ram running Windows Vista operating system, and Matlab 7.1. To make the approach easily repeatable, we used *Prtools Toolbox* for classification methods [19].

4.1. Case Study. An expert is first asked for the parameters Δ_x and Δ_y. Those parameters are selected as $\Delta_x = 0.12$ and $\Delta_y = 0.07$ by the expert. Using those parameters, activity and inactivity maps are created as in Figures 7 and 8, respectively. Using those activity and inactivity maps, the dataset is filtered using a threshold value, %40. This threshold value is decided by the expert in order to keep more information unfiltered. Although it may seem that using a 2D activity maps rather than a 3D representation may cause loss of data. It is important to note that these maps are only used in finding approximate locations of clusters where unfiltered data and the preliminary information about clusters are fully preserved. Lastly, using unfiltered data and preliminary information about the clusters, the system finds final active fragments. To visualize 3D topology of active fragments, an active template molecule that is selected by the expert is used. Graph-based representation of extracted active fragments on the template molecule is shown at Figure 9.

We measure the processing time of each step of our approach during the case study. We tabulate these values in Table 2, where the time consumed during human-computer interactions is neglected. The table shows that the most costly part of our approach is clustering. Our approach consumes around 20 seconds for clustering, while it consumes around only 2, 0.5, and 0.7 seconds for activity map creation, filtering, and fragment extraction, respectively. The total time consumed by the stages of the approach is around 23 seconds. The overall time from beginning to finding active fragments on the case study (including the time consumed during human-computer interactions) is around 11 minutes.

4.2. Benchmark Comparisons: Antituberculosis Dataset. In this section, using the antituberculosis dataset, we compare our approach with two well-known approaches from literature: SUBDUE and FSG. These approaches are chosen, because they are often used in literature to find frequent

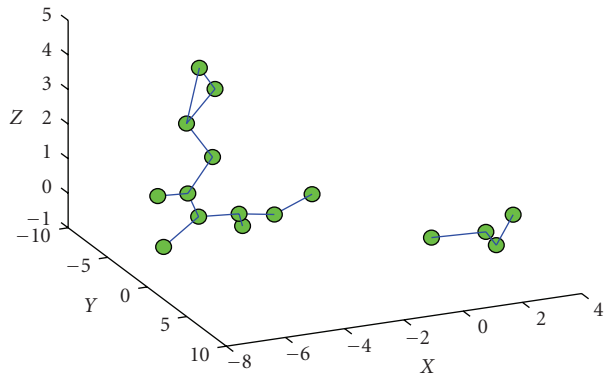

Figure 9: Active fragments.

Table 2: Processing time for each step in the case study (in seconds).

Activity map creation	2.045
Filtering	0.513
Clustering	19.876
Fragment extraction	0.728
Total	23.162

Table 3: Performance of the proposed approach, SUBDUE and FSG with respect to recall and precision metrics.

	Proposed approach	SUBDUE	FSG
Recall	0.95	0.80	0.40
Precision	0.97	0.75	0.67

substructures of graphs and molecules. Like our approach, SUBDUE uses molecular graphs labeled as active or inactive. However, FSG cannot use labeled graphs. Therefore, we have used only the active molecules to determine active fragments by FSG and neglected inactive molecules which cause a great decrease in the success of FSG. In our experiments, we use publicly available original implementations of SUBDUE (http://ailab.wsu.edu/subdue) and FSG (http://glaros.dtc.umn.edu/gkhome/pafi/overview) in order to increase reliability and repeatability. For the basis of comparisons, we use a set of active fragments (F_A^*) that are determined for antituberculosis using extensive analysis in [7].

In order to compare SUBDUE, FSG, and the proposed approach, first we compute active fragments for antituberculosis dataset using each of these approaches. Let F_A^{subdue}, F_A^{fsg}, and F_A^P represent the sets of active fragments computed by SUBDUE, FSG, and the proposed approach, respectively. Then, we compare the most significant fragments in F_A^{subdue}, F_A^{fsg}, and F_A^P with the ones in F_A^*. By the most significant fragment, we mean the fragment that has the largest support value, which represents the percentage of active molecules including this active fragment at the dataset. Let S_A^{subdue}, S_A^{fsg}, S_A^P, and S_A^* denote the most significant fragments in F_A^{subdue}, F_A^{fsg}, F_A^P, and F_A^*.

For comparison of the methods, we used two measures: *recall* and *precision*. Recall is defined in (2) as the ratio of correctly discovered bonds of S_A^* by an approach. In the equation, $f(b, S)$ is a function that returns 1 if the bond b is contained by the fragment S ($b \in S$); otherwise, it returns 0. High recall value of an approach implies that it can correctly find most of the bonds in S_A^*, which is the most significant active fragment

$$\text{recall}(X) = \frac{\sum_{b \in S_A^*} f\left(b, S_A^X\right)}{\left|S_A^*\right|}. \tag{2}$$

Using only the recall metric, we cannot measure success of an approach in determining active fragments, because this measure does not use the excess bonds found by the approach. That is, recall of an approach X is 1 if $S_A^X \equiv S_A^*$ or $S_A^* \in S_A^X$, where X finds also the bonds that are not a part of an active fragment (S_A^*). Therefore, we introduce precision metric in (3). The precision value of an approach X is high if all of the bonds in S_A^X are included in S_A^*. If both of the recall and precision values are close to 1.0 for an approach, this means that this approach can correctly and precisely find most significant active fragments

$$\text{precision}(X) = \frac{\sum_{b \in S_A^X} f\left(b, S_A^*\right)}{\left|S_A^X\right|}. \tag{3}$$

The results of our experiments are shown at Table 3. The table implies that our approach can correctly determine the most significant active fragments, because its recall and precision values are both close to 1.0. Recall and precision values of SUBBDUE are 0.8 and 0.75, respectively. Although the performance of SUBDUE is also high, our approach significantly outperforms it. Unlike SUBDUE and the proposed approach, the performance of FSG is low; its recall and precision values are 0.40 and 0.67, respectively. This performance difference is intuitive, because FSG cannot use the class information (e.g., active molecule and inactive molecule) while determining fragments. Although FSG can find fragments belonging to active molecules, these fragments may not be active fragments, because they may also repeat in the structure of inactive molecules. However, unlike FSG but similar to the proposed approach, SUBDUE uses class information while determining frequent fragments. Hence, it can determine frequent fragments that exist in active molecules but not in inactive molecules. Therefore, in our experiments, performance of SUBDUE is higher than the performance of FSG. These findings imply that inactive molecules may have a significant importance on determining active fragments. Hence an approach that uses both active and inactive molecules should be used to solve active fragment discovery for drug design.

4.3. Benchmark Comparisons: Synthetic Datasets. One of the main deficiencies in the methods proposed in graph-based data mining is their narrow view on the problem. These methods search for the common fragments that repeat exactly the same in molecules. However, molecules may have fragments that give them activity with respect to a specific disease, but these fragments may not repeat exactly in each

active molecule. Instead, these fragments may repeat with some deviations, which implies the necessity of subgraph mining methods that discover not only exactly repeating substructures but also the ones that repeat with some deviations [15]. In this paper, we propose a clustering-based molecular graph mining method for frequent substructure extraction, where pieces (atom-bond-atom triples) with similar features fall into the same clusters. This enables frequent substructures to be discovered even though these substructures repeat in molecules with some variations.

In this part of our experiments, using synthetic datasets, we show how successful our approach is with respect to others when substructures giving activity to a molecule do not exactly repeat but repeat with some small variations. We compare the proposed approach with ParMol (http://www2.cs.fau.de/Forschung/Projekte/ParMol/) package for frequent molecular subgraph mining [20], which includes publicly available implementations of gSpan [13], Gaston [17], MoFa [3], and FFSM [18]. We compare our approach with these well-known methods using the synthetic datasets. Like the proposed approach, ParMol package enables molecular mining using graphs labeled as active and inactive.

A synthetic graph dataset is composed of two sets of graphs: the graphs representing active molecules (S_a) and the graphs representing inactive molecules (S_i). In these graphs, labels are real numbers, where the nodes and edges take their labels in the ranges of [0–4] and [0–6], respectively. Before creating the graphs in S_a or S_i, we first create three different sets of patterns: the subgraphs repeating only in active molecules (P_a), the subgraphs repeating only in inactive molecules (P_i), and lastly the subgraphs repeating both in active and inactive molecules (P_{ai}). These three sets of patterns are inspired by the real-life datasets [7]. Each graph $G \in S_a$ is created so that it includes subgraphs from both P_a and P_{ai}. However, before adding these subgraphs to G, we modify them according to a parameter p_n, called probability of noise. This is done because of the fact that a pattern may not exactly repeat in real-life molecular datasets; instead the same pattern may repeat in different molecules with slight variations [15].

If $p_n = 0$, then we directly add subgraphs from S_a and S_{ai} to G. However, if $p_n > 0$, then we slightly modify the labels of the nodes and edges in these subgraphs with probability p_n by adding some noise, before adding them to G. The amount of noise to be added is determined randomly in the range of [0–0.25]. We also add m other nodes to G, where the number m and the labels of these nodes are chosen randomly. In order to ensure connectivity of the graph, we add edges randomly between these nodes and the others. Lastly, we randomly give a unique ID to each node in the graph. Similarly, we create graphs for inactive molecules, but this time these graphs are produced using the patterns from P_i and P_{ai}. Using this procedure, we compute a number of graphs representing active and inactive molecules. In the resulting graph dataset, some similar patterns repeat in almost every graph, while some others repeat only in the graphs representing either the active molecules or the inactive molecules. There is a similar case in the real-life datasets, where only the substructures repeating in active molecules are responsible for the activity of the molecule, while the substructures repeating almost in every molecule or only in the inactive molecules do not have any significant effect on the activity.

We created eight different datasets using different numbers of active and inactive molecules as well as different values for the probability of noise. Each graph in our dataset has 20 nodes and 40 edges on the average. For each dataset, we exactly know which subgraphs are repeating only in the graphs representing active molecules. Note that these subgraphs are not repeating exactly the same but with some small differences (called noise). Hence, we can quantitatively measure how successful our approach is, compared to the other approaches (*gSpan*, *Gaston*, *MoFa*, and *FFSM*), while finding these subgraphs. Table 4 summarizes our results for competing subgraph mining methods on our datasets.

Our experiments show that the proposed approach and the graph mining methods *gSpan*, *Gaston*, *MoFa*, and *FFSM* can find all of the active substructures correctly when there is no noise ($p_n = 0$). However, an increase in the probability of noise results in a dramatic performance decrease in the graph mining methods *gSpan*, *Gaston*, *MoFa*, and *FFSM*. For example, when p_n is increased to 0.25, although their precisions remain 1.0, the recall values of *gSpan*, *Gaston*, *MoFa*, and *FFSM* decrease to 0.33, 0.28, 0.21, and 0.19, respectively. This means that they do not find fragments that are not frequent (precision is 1.0), but they can only find 33%, 28%, 21%, and 19% of the frequent fragments, respectively. On the other hand, precision and recall of the proposed approach are both 1.0, which means that the proposed approach can correctly determine frequent fragments even though these fragments do not eactly repeat in molecules, but repeat with some variations ($p_n = 0.25$).

The situation becomes more dramatic when p_n becomes 1.0; both precision and recall become zero for *gSpan*, *Gaston*, *MoFa*, and *FFSM*. This means that they cannot find any part of the frequent fragments, because nodes and edges in these fragments do not exactly repeat in molecules. In all these cases, our approach can find almost all of the active substructures correctly; it has precision and recall values almost equal to 1.0 when $p_n > 0$. For varying size of the datasets, performance of our approach does not change. These experiments show that our approach can successfully find frequent substructures even though these substructures do not repeat exactly in the molecules; however, the other methods can find these substructures correctly only if they repeat exactly the same in the molecules ($p_n = 0$).

4.4. Using Fragments for Classification. Searching a molecular substructure rapidly in a molecular database is an important research problem in drug design. In literature, graph-mining techniques are mostly used to search molecular databases for the new molecules that are likely to be active for a specific disease [1, 21]. Using classical graph-searching methods, it is difficult to find molecules with specific frequent substructures, because of the time complexity of subgraph isomorphism. Instead of using graph-searching methods, classification methods from machine learning literature are also used for the estimation of active and inactive molecules

Table 4: Experimental results for the synthetic datasets (P refers to *precision* and R refers to *recall*).

Dataset number	Number of active molecules	Number of inactive molecules	Probability of noise [0-1]	Proposed method		gSpan		Gaston		MoFa		FFSM	
				P	R	P	R	P	R	P	R	P	R
1	10	10	0.0	1	1	1	1	1	1	1	1	1	1
2	10	10	0.25	1	1	1	0.33	1	0.28	1	0.21	1	0.19
3	10	10	0.5	1	1	1	0.12	1	0.11	1	0.08	1	0.06
4	10	10	1.0	1	0.91	0	0	0	0	0	0	0	0
5	50	50	0.0	1	1	1	1	1	1	1	1	1	1
6	50	50	1.0	1	0.94	0	0	0	0	0	0	0	0
7	100	100	0.0	1	1	1	1	1	1	1	1	1	1
8	100	100	1.0	1	0.95	0	0	1	0.05	0	0	0	0

Table 5: Performance of the classification methods on the *antituberculosis* dataset.

	Unfiltered data		Filtered data	
	Active molecules	Inactive molecules	Active molecules	Inactive molecules
Classifiers	Success (%)	Success (%)	Success (%)	Success (%)
LDA	0	100	92	95
1-NN	100	100	100	95
SVM	100	100	100	100
DT	100	100	100	100
Average processing time	39 min. and 19 sec.		14 min. and 29 sec.	

in a molecular database. Classification methods require each instance in a training set to have the same dimensions. Therefore, the molecular data should be preprocessed before using these methods.

In this paper, we propose an approach for deriving active fragments that account for the activity of the active molecules. We can also derive inactive fragments from inactive molecules using the same method. In this case, the derived inactive fragments account for the inactivity of the inactive molecules. Then, we can use information about activity and inactivity clusters (representing active and inactive fragments) as features to represent each molecule using the same dimensions. Let us have n_1 activity clusters and n_2 inactivity clusters that we derive using the proposed approach. For each molecule, we prepare an array of $n_1 + n_2$ dimensions, where each dimension represents one cluster. If the molecule has an active or inactive fragment, the corresponding dimension of the vector is set to the minimum distance from the points of molecules to the corresponding cluster's center; otherwise, it is set to 0. This way, we represent molecules using vectors of the same dimensions. Using this methodology, we create a training set from the labeled examples. Then, we input this training set to different classifiers: LDA, k-NN, SVM, and decision trees (DT). Lastly, using one-leave-out cross validation method, we measure the classification performance of the classifiers.

A similar approach is used by Macaev et al. in [7]. They first find a active and b inactive fragments for antituberculosis dataset. Then, for each molecule in the dataset, they compose a feature array of $a + b$ dimensions, where each dimension takes binary values (1 or 0) to show whether a specific fragment exists in the molecule or not. Using the derived arrays as training set, they train a classifier to measure how well the derived fragments can be used to classify molecules as active or inactive with respect to antituberculosis. Their activity/inactivity prediction using leave-one-out cross validation is 80%.

In order to measure the performance of our approach better, we also train the classifiers with the original unfiltered data that contain not only active/inactive fragments but also other fragments that are filtered out while determining these active/inactive fragments. Unfiltered data contain more information about the molecules, so classifiers using the unfiltered data may have a better performance with respect to their performance using only a subset of these data (i.e., only active and inactive fragments). Our main aim in this paper is to correctly determine active/inactive fragments that are responsible for the activity/inactivity of the molecules. If we determine these fragments correctly, the classifiers using only those fragments as features should also demonstrate a good performance in classifying molecules.

In Table 5, we tabulate our results for $\Delta_x = 0.12$ and $\Delta_y = 0.07$ using the original (unfiltered) data and the filtered data, where only active and inactive fragments are used as features. Our results show that most of the classifiers (except LDA) achieve the same performance when they use only active/inactive fragments as features (filtered data) and when they use the whole molecular data (unfiltered data). In our

experiments, the best performance belongs to SVM and DT classifiers, which always correctly classify active and inactive molecules (success is 100%). Performance of other classifiers is also very good; almost all of the classifiers can correctly classify molecules in more than 95% of the cases.

We immediately recognize that LDA cannot find active molecules and labels all of the molecules as inactive when unfiltered data are used. The reason behind this is the fact that LDA maps all data into a $(C - 1)$-dimensional feature space, where C is the number of classes [16]. In our case, there are only two classes, which means that LDA tries to discriminate these two classes in a 1-dimensional space. Since the unfiltered data contain highly overlapping structures and features between active and inactive molecules, classification in a 1-dimensional space is clearly not enough when unfiltered data are used. However, when we filter the data using the proposed method, we remove the common and noisy parts of the molecules that are not significant for being active or inactive. Hence, LDA can successfully learn active and inactive molecules in a 1-dimensional space using the filtered data.

Those results imply that our approach can correctly determine the active and inactive fragments, and those fragments can successfully be used as features in classification. Moreover, when the classifiers use only the active/inactive fragments extracted from filtered data rather than fragments extracted from original (unfiltered) data, it is observed that overall classification process is 2.7 times faster.

5. Conclusions

Infectious diseases like tuberculosis are the leading causes of death and suffering worldwide. The World Organization of Health reports a significant rise in drug resistance of diseases like tuberculosis [22]. This increasing drug resistance highlights the need for new, safer, and more effective drugs. However, designing new drugs is an exhaustive process, where discovery of fragments that account for biological activity and prediction of biological activity in relation to the chemical structures of molecules are crucial.

In this paper, we extend our previous studies in [23]. Unlike the previous work, this paper discusses the role of human experts in the process, presents computational complexity of the overall approach, analyzes the robustness of the approach to the noise, and lastly compares the proposed approach with current approaches FFSM, MoFa, gSpan, and Gaston in detail using synthetic datasets. Our work has two main contributions. First, previous graph-based approaches for frequent pattern mining methods are fully automated and do not allow experts to interact with the system to incorporate their expertise to the process. However, the proposed approach enables experts to interact with the system and improve the solution with their expertise. Second, current methods can determine frequent patterns only if these patterns exactly repeat in molecules. However, in many settings, patterns may not repeat exactly, but with some variations. For example, in different conditions (e.g., temperatures), some features of the bonds and atoms in a molecule may slightly change (e.g., orientation of atoms); this means that the same pattern may appear slightly different even in the same molecule under different conditions. While current methods may not determine these patters that appear with slight variation in molecules, our approach successfully determines them using clustering.

We have evaluated the performance of our approach using experiments on antituberculosis dataset and various synthetic datasets. Our experiments show that our approach can correctly determine active fragments of molecules that account for the activity of those molecules. We also show using our approach that classification methods can achieve good performances while determining biological activity or inactivity in relation to the chemical structures of molecules.

In this work, we use antituberculosis dataset in order to demonstrate and evaluate our approach. As a future work, we want to evaluate our approach using the dataset for other diseases as well.

References

[1] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1036–1050, 2005.

[2] I. Fischer and T. Meinl, "Graph based molecular data mining—an overview," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, P. Wieringa, M. Pantic, and M. Ludema, Eds., pp. 4578–4582, IEEE Computer Society, The Hague, The Netherlands, October 2004.

[3] C. Borgelt and M. R. Berthold, "Mining molecular fragments: finding relevant substructures of molecules," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '02)*, V. Kumar, S. Tsumoto, N. Zhong, P. Yu, and X. Wu, Eds., pp. 51–58, IEEE Computer Society, Maebashi City, Japan, 2002.

[4] G. Yang, "Computational aspects of mining maximal frequent patterns," *Theoretical Computer Science*, vol. 362, no. 1–3, pp. 63–85, 2006.

[5] D. J. Cook and L. B. Holder, "Graph-based data mining," *IEEE Intelligent Systems and Their Applications*, vol. 15, no. 2, pp. 32–41, 2000.

[6] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '01)*, N. Cercone, T. Y. Lin, and X. Wu, Eds., pp. 313–320, IEEE Computer Society, San Jose, Calif, USA, November-December 2001.

[7] F. Macaev, G. Rusu, S. Pogrebnoi, et al., "Synthesis of novel 5-aryl-2-thio-1,3,4-oxadiazoles and the study of their structure-anti-mycobacterial activities," *Bioorganic and Medicinal Chemistry*, vol. 13, no. 16, pp. 4842–4850, 2005.

[8] A. Nayyar, V. Monga, A. Malde, E. Coutinho, and R. Jain, "Synthesis, anti-tuberculosis activity, and 3D-QSAR study of 4-(adamantan-1-yl)-2-substituted quinolines," *Bioorganic and Medicinal Chemistry*, vol. 15, no. 2, pp. 626–640, 2007.

[9] M. Murcia-Soler, F. Pérez-Giménez, F. J. García-March, M. T. Salabert-Salvador, W. Díaz-Villanueva, and P. Medina-Casamayor, "Discrimination and selection of new potential antibacterial compounds using simple topological descriptors," *Journal of Molecular Graphics and Modelling*, vol. 21, no. 5, pp. 375–390, 2003.

[10] A. An and Y. Wang, "Comparisons of classification methods for screening potential compounds," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '01)*, N. Cercone, T. Y. Lin, and X. Wu, Eds., pp. 11–18, IEEE Computer Society, San Jose, Calif, USA, November-December 2001.

[11] T. Washio and H. Motoda, "State of the art of graph-based data mining," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 1, pp. 59–68, 2003.

[12] A. Inokuchi, T. Washio, and H. Motoda, "An aprioribased algorithm for mining frequent substructures from graph data," in *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, D. A. Zighed, H. J. Komorowski, and J. M. Zytkow, Eds., pp. 13–23, Springer, Lyon, France, September 2000.

[13] X. Yan and J. Han, "gSpan: graph-based substructure pattern mining," in *Proceedings of the IEEE International Conference on Data Mining (ICDM '02)*, V. Kumar, S. Tsumoto, N. Zhong, P. Yu, and X. Wu, Eds., pp. 721–724, IEEE Computer Society, Maebashi City, Japan, December 2002.

[14] M. J. E. Sternberg and S. H. Muggleton, "Structure activity relationships (SAR) and pharmacophore discovery using inductive logic programming (ILP)," *QSAR and Combinatorial Science*, vol. 22, no. 5, pp. 527–532, 2003.

[15] N. Shvets and A. S. Dimoglo, "The electron-topological method (etm): its further development and use in the problems of SAR study," in *Molecular Modeling and Prediction of Bioactivity*, K. Gundertofte and F. S. Jorgensen, Eds., pp. 418–429, Kluwer Academic/Plenum Publishers, New York, NY, USA, 1999.

[16] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, Cambridge, Mass, USA, 2001.

[17] S. Nijssen and J. N. Kok, "A quickstart in frequent structure mining can make a difference," in *Proceedings of the 10 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pp. 647–652, 2004.

[18] J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pp. 549–552, 2003.

[19] F. van der Heijden, R. P. Duin, D. de Ridder, and D. M. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using Matlab*, John Wiley & Sons, New York, NY, USA, 2004.

[20] T. Meinl, M. Wörlein, O. Urzova, I. Fischer, and M. Philippsen, "The parmol package for frequent subgraph mining," *Electronic Communications of the EASST*, vol. 1, pp. 1–12, 2007.

[21] S. Nijssen and J. N. Kok, "Frequent graph mining and its application to molecular databases," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, P. Wieringa, M. Pantic, and M. Ludema, Eds., vol. 5, pp. 4571–4577, IEEE Computer Society, The Hague, The Netherlands, October 2004.

[22] M. Rich, *Guidelines for the Programmatic Management of Drug-Resistant Tuberculosis*, World Health Organization, Geneva, Switzerland, 2006.

[23] B. Yılmaz, M. Göktürk, and N. Shvets, "User assisted substructure extraction in molecular data mining," in *Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*, P. Perner and O. Salvetti, Eds., vol. 5108 of *Lecture Notes in Artificial Intelligence*, pp. 12–26, Springer, New York, NY, USA, 2008.