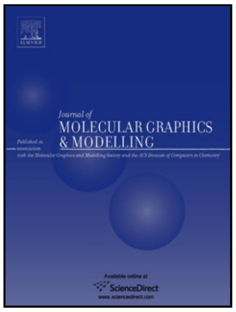


Title: Predicting Activity Approach based on New Atoms SimilarityKernel Function

Author: Ahmed H. Abu El-Atta M.I. Moussa Abul Ella Hassenian

PII: S1093-3263(15)30006-1
DOI: http://dx.doi.org/doi:10.1016/j.jmgm.2015.05.014
Reference: JMG 6554

To appear in: *Journal of Molecular Graphics and Modelling*

Received date: 24-12-2014
Revised date: 28-3-2015
Accepted date: 26-5-2015

Please cite this article as: Ahmed H. Abu El-Atta, M.I. Moussa, Abul Ella Hassenian, Predicting Activity Approach based on New Atoms SimilarityKernel Function, *Journal of Molecular Graphics and Modelling* (2015), http://dx.doi.org/10.1016/j.jmgm.2015.05.014

Predicting Activity Approach based on New Atoms Similarity Kernel Function

Ahmed H. Abu El-Atta [1,2], M. I. Moussa [2] and Abul Ella Hassenian [1,3]

[1] *Scientific Research Group in Egypt (SRGE), egyptscience.net, Egypt*
[2] *Faculty of Computers and Information, Benha University, Benha, Egypt*
[3] *Faculty of Computers and Information, Cairo University, Egypt*
ahmed.aboalatah@fci.bu.edu.eg (Ahmed H. Abu El-Atta)
mahmoud.mossa@fci.bu.edu.eg (M. I. Moussa)
abo@egyptscience.net (Abul Ella Hassenian)

Abstract

Drug design is a high cost and long term process. To reduce time and costs for drugs discoveries, new techniques are needed. Chemoinformatics field implements the informational techniques and computer science like machine learning and graph theory to discover the chemical compounds properties, such as toxicity or biological activity. This is done through analyzing their molecular structure(molecular graph). To overcome this problem there is an increasing need for algorithms to analyze and classify graph data to predict the activity of molecules. Kernels methods provide a powerful framework which combines machine learning with graph theory techniques. These kernels methods have led to impressive performance results in many several chemoinformatics problems like biological activity prediction. This paper presents a new approach based on kernel functions to solve activity prediction problem for chemical compounds. First we encode all atoms depending on their neighbors then we use these codes to find a relationship between those atoms each other. Then we use relation between different atoms to find similarity between chemical compounds. The proposed approach was compared with many other classification methods and the results show competitive accuracy with these methods.

Keywords: Chemoinformatics, Graph kernel, Machine learning, Activity prediction, Drug discovery

1. Introduction

Chemoinformatics (chemical informatics) is the field that seeks to use informational techniques like computer science, mathematics and information techniques to predict or analyze molecule's (chemical compounds) properties. One of the major principles in this research field is the similarity principle, which states that two structurally similar molecules should have similar activities and properties.

Graphs are flexible models that have been used to present data in many scientific, engineering, and business fields. For example, in finance data analysis, graphs are used to model dynamic stock price changes (Jin *et al*., 2007). To analyze biological data, graphs have been utilized in modeling chemical structures (Smalter *et al*., 2008), protein sequences (Weston *et al*., 2006), protein structures (Huan *et al*., 2004), and gene regulation networks (Huang *et al*., 2007). The structure of a molecule is encoded by a labeled graph $G = (V, E, \mu, \pi)$, where the unlabeled graph (V, E) encodes the structure of the molecule while μ maps each vertex to an atom's label and π maps each edge to a type of bond between two atoms (single, double, triple or aromatic).

Presenting chemical compounds by graphs enables us to apply graph classification techniques to predict chemical compounds properties. In graph classification, each graph is associated with a target value and the aim is to find a good function that maps graphs to their target values. The existing algorithms of classifying graph data can be divided into three categories (Poezevara *et al*., 2009; Brun *et al*., 2010).

The first approach is introduced within the Quantitative Structure Activity Relationship (QSAR) field which is based on finding the correlation between molecule's descriptors and molecule's properties (Abu El-Atta *et al*., 2014). Vectors of molecular descriptors (MDs) may be defined from structural information (Cherqaoui and Villemin, 1994) besides physical properties or biological activities. Molecular descriptors may be classified into three categories. The first is simple one-dimensional (1D) descriptors which represent bulk properties of compounds, the second is two-dimensional (2D) descriptors such as topological and charge indices, and the last is complex three-dimensional (3D) descriptors which often rely on 3D representation and conformational aspects of a molecule (Turner *et al*., 2004). Molecular descriptors may be used within any statistical machine learning algorithm to predict molecule's properties. Such a scheme allows benefiting from the large set of tools available within the statistical machine learning framework.

Another approach is to explicitly collect a set of features from the graphs. Features may be chosen from paths, cycles, trees, and subgraphs. Once a set of features is determined, a graph is described by a feature vector. With a collection of vectorized graph data, any existing data mining method that works in n-dimensional euclidean space may be applied to do graph classification. In the context of chemoinformatics, explicit pattern features for compounds are often known as structural keys. A structural key is a bit string denoting presence of certain patterns (such as paths, cycles, trees, etc.) of interest (Smalter *et al.*, 2010).

The third approach of graph classification is to implicitly collect a set of features (possibly an infinite number of such features) and compute the similarity of two graphs via a kernel function. The term kernel function refers to an operation of computing the inner product between two points in a Hilbert space, so this may lead to avoiding the explicit computation of coordinates in that feature space. Graph kernel functions are simply kernel functions that have been defined to compute the inner product between two graphs.

Many of graph kernel functions have been developed, with promising application results as described in (Ralaivola *et al.*, 2005). Among these methods, some kernel functions draw on graph features such as walks (Kashima *et al.*, 2003) or cycles (Horvath *et al.*, 2004), while others may use different approaches such as genetic algorithms (Barbu *et al.*, 2006), frequent subgraphs (Deshpande *et al.*, 2005), or graph alignment (Fröhlich *et al.*, 2006). In (Grenier *et al.*, 2014), a local subgraph is used to encode the stereo-isomerism property of each atom of a molecule. A kernel between bags of such subgraphs provides a similarity measure incorporating stereo-isomerism properties.

In this paper, we present an approach based on the similarity principle to classify chemical compounds. Chemical compounds with similar properties have similar structure and compounds with similar structure contain common subgraphs. How an atom is connected to its neighbors, plays a great role to determine the properties of that atom. Based on this information we design an approach in which chemical graphs are not explicitly factored into patterns but only the count of these patterns is used.

The proposed approach starts by collecting all atoms in a given dataset then encoding them by unique codes. We use atoms codes to find similarity relationship between atoms by counting the common subgraphs between each two atoms and their neighbors. After that, the relationship between atoms is used to create a similarity vectors between compounds each other. Finally the kernel function is applied to similarity vectors to predict the activity of compounds. The rest of this paper is organized as the following in section 2, we discuss the proposed approach and state its steps. In section 3, we describe the atom coding system and how to employ it to find similarity between atoms. In section 4, we describe how to compute similarity between chemical compounds and the kernel function. In section 5, we present our results and performance evaluation. Conclusion to our work is presented in section 6.

2. Basic Steps of The Prediction Approach

Activity of an atom in a chemical compound depends on the bonds which connect this atom with its neighbors. We present coding system to represent each atom. This coding system preserves the atom type, types of all adjacent atoms and bonds types between that atom and its neighbors. As Figure 1 shows, we firstly use this coding system to encode all atoms in all chemical compounds of a given dataset. This coding system enables us to sort all atoms according to that code to create an index for them.

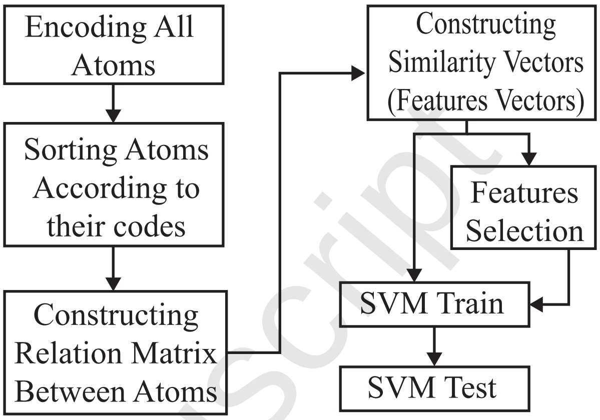

Encoding All Atoms
Sorting Atoms According to their codes
Constructing Relation Matrix Between Atoms
Constructing Similarity Vectors (Features Vectors)
Features Selection
SVM Train
SVM Test

Figure 1: Steps of the prediction approach.

Moreover, from this code we construct a relation between these atoms each other. Finally, the relation matrix between atoms is used to find a similarity matrix between chemical compounds each other. Rows of this similarity matrix represent the features vectors for each compound. These features vectors may be directly passed to the kernel function to create the kernel matrix for dataset or a features selection based on ranking is done before using kernel function. Steps of finding atoms similarity are described in the following section.

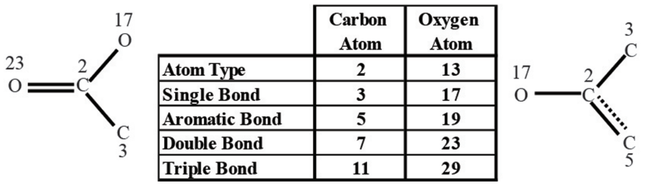
3. Atoms Similarity

3.1. Atoms coding system

Any composite number is the result of multiplication of at least two prime numbers and any prime number is only divisible by one or itself. From the previous facts we build a coding system depending on prime numbers to represent each atom and its incident bonds and adjacent atoms. Prime numbers were used previously in many algorithms. An algorithm proposed by (Weininger *et al.*, 1989) uses prime numbers to improve performance of Morgan algorithm (Morgan, 1965) and to create canonical SMILES. Morgan's algorithm proceeds in two steps. In the first step, each atom is labeled by the number of neighbors of that atom (atom degree). In the second step, atoms labels are calculated in a recursive manner. At each recursive the label of any given atom is the sum of the labels of its neighbors computed in the previous step. The main criticism of Morgan's algorithm is the ambiguity of the summation when computing atom labels.

To overcome this problem, (Weininger *et al.*, 1989) proposed a solution using prime numbers. In this implementation the initial labels are substituted by primes, that is, degree 1 is replaced by 2, degree 2 by 3, degree 3 by 5, and so on. Next, instead of summing the labels of the neighbors, the product of the primes is computed. According to the prime factorization theorem, the solution of (Weininger *et al.*, 1989) is unambiguous. Also, prime numbers are used as labels for bonds to differ between bonds types. Prime ID number is a modification of the Randić connectivity ID number, which aimed to improving the discriminating power of Randić connectivity ID number. In practice, prime ID numbers are calculated by substituting the edge connectivity of the connectivity ID number with a different edge weight based on the first nine prime numbers (Todeschini *et al.*, 2009).

The proposed coding system combines the benefits of using prime numbers as labels for atoms and bonds. Figure 2, shows two atoms and their neighbors. The one on the left is carbon atom placed in the center and it has two single bonds with two other atoms, another carbon atom and an oxygen atom, also it has double bond with another oxygen atom. On the right a carbon atom placed in the center and it has two single bonds with two other atoms, another carbon atom and an oxygen atom, also it has an aromatic bond with another carbon atom. These atoms and their neighbors represent a class of graphs called star graphs; each star graph has one central atom.



	Carbon Atom	Oxygen Atom
Atom Type	2	13
Single Bond	3	17
Aromatic Bond	5	19
Double Bond	7	23
Triple Bond	11	29

Figure 2: Two atoms and their neighbors.

To encode each atom we create a table with 5 rows and 118 columns as the atoms can be labeled by each of the 118 chemical elements. The numbers in the first row represent central atom type and the numbers in each row of the other four rows represent one type of the chemical bonds (single, aromatic, double, and triple bond). Numbers in that table are prime numbers, an example of that table is shown in Figure 2. The left atom in Figure 2 is encoded as the following, first the central atom is a carbon atom and according to the table in Figure 2 it is represented by prime number 2, and it is adjacent to two oxygen atoms. The first one is connected to the carbon atom by a single bond and according to the table it is represented by 17. The other one has double bond and according to the table it is represented by 23. Also, the central carbon atom is adjacent to another carbon atom with single bond and according to the table it is represented by 3. After mapping each atom to a corresponding prime number from the given table, the product of these prime numbers represents the code of the atom which equals 2346 and it can be computed by Equation (1). As the same way the right carbon atom in Figure 2 is encoded as the product of 2 * 3 * 5 * 17 which equals to 510.

$$Code(A_i) = \prod_{l=1}^{d} q_l \tag{1}$$

Where A_i is an atom, d is the degree of the atom, and q_l is the prime number corresponding to the l^{th} edge incident to A_i.

For any two atoms have the same code, it is clear that they must be of the same type and have the same neighbors. These atoms not only have the same neighbors but also they are connected to their neighbors by the same types of chemical bonds. For example two carbon atoms in two stars in Figure 3 both of them have atom code equals 6 and they are identical. On the other hand if two atoms are different in any part of their structure (e.g. their types, one of their neighbors, or one of bonds types) they will have different codes. For example two oxygen atoms in the top star in Figure 3, their codes are 39 and 91. Their codes are not identical because each of them is connected to the carbon atom with a different type of chemical bonds. This code system allows us to encode each star subgraph in any chemical compound by a number which preserves its structure. During reading chemical compounds of a given dataset all star subgraphs can be encoded and inserted in a binary tree structure.

$$Com(A_i, A_j) = \left| \left\{ q_1, q_2, \cdots, q_n : \prod_{l=1}^{n} q_l = gcd(A_i, A_j) \right\} \right| \tag{2}$$

Also this code can be used to find the number of edges in the common subgraph between two stars. Let $\prod_{l=1}^{n} q_l$ is the greatest common divisor between codes of two stars atoms A_i and A_j and q_l is a prime number for all $l = 1, 2, \cdots, n$. Then the number of edges in the common subgraph between A_i and A_j can be computed from the number of primes that compose their greatest common divisor. The number of primes that compose the greatest common divisor between A_i and A_j can be computed by Equation (2). Equation (3) represents the relationship between the greatest common divisor for codes which represent two stars A_i and A_j and the number of edges in the common subgraph between these two stars.

$$Com_e(A_i, A_j) = \begin{cases} Com(A_i, A_j) - 1 & \text{if } Com(A_i, A_j) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

3.2. Atoms Common Subgraphs Matrix (Atoms Relationship Matrix)

After encoding, sorting and indexing all star subgraphs in all dataset we construct the atoms common subgraphs matrix S (atoms relationship matrix) between the atoms each other. Matrix S is a squared matrix which its size equals to the number of unique star subgraphs in dataset (Training set). Each element at position i, j in S matrix represents the number of common connected subgraphs between stars subgraphs A_i and A_j which their order in the sorted stars is i and j. Since each edge is retrieved from its two extremities, each edge and each atom are counted twice. We avoid this problem by counting each edge as $\frac{1}{2}$ times and only central atom is counted as subgraph. So each element of S matrix can be defined by Equation (4).

$$S_{ij} = \begin{cases} 1 + \frac{n}{2} + \sum_{i=2}^{n} \binom{n}{i} & \text{if are the same atom and } n > 0, \\ 1 & \text{if are the same atom and } n = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Where n is the number of common edges between stars A_i and A_j. Number of common edges between stars A_i and A_j can be computed by Equation (3). The value 1 in the first case of the Equation (4) corresponds to the central atom. The second term in the equation which equals $\frac{n}{2}$ corresponds to the number of common edges. The number of edges is divided by two to avoid counting them twice. The third term in the equation is $\sum_{i=2}^{n}\binom{n}{i}$. Each part in this summation corresponds to the number of subgraphs which contains $2, 3, 4$, and up to n edges.

Central Atom	2	2	13	13	2	2
Edges	3	5	3	7	3, 5, 17	3, 17, 23
Atom Code	6	10	39	91	510	2346
6	n=1 1.5	n=0 1	n=1 0	n=0 0	n=1 1.5	n=1 1.5
10	n=0 1	n=1 1.5	n=0 0	n=0 0	n=1 1.5	n=0 1
39	n=1 0	n=0 0	n=1 1.5	n=0 1	n=1 0	n=1 0
91	n=0 0	n=0 0	n=0 1	n=1 1.5	n=0 0	n=0 0
510	n=1 1.5	n=1 1.5	n=1 0	n=0 0	n=3 6.5	n=2 3
2346	n=1 1.5	n=0 1	n=1 0	n=0 0	n=2 3	n=3 6.5

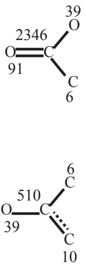


Figure 3: The atoms common subgraphs matrix S between two chemical compounds in the right side of the figure.

Number of the common subgraphs between two stars in Figure 3 is computed with the following steps. First, we count the number of common edges between two stars which is 2 edges. The first edge is the single bond between two carbon atoms and the second edge is the single bond between central carbon atom and oxygen atom. These edges can be determined from the codes of these stars by finding the common primes between codes. In this case we apply the first part of Equation (4) because the number of common edges is greater than 0. These two stars have four common subgraphs. The first one is the central carbon atom. The other two subgraphs are the two edges, each of them is counted as $\frac{1}{2}$. The last common subgraph is the subgraph which consists of the two edges. From the previous it is clear that the result of the equation equals three. Also Figure 3 shows that the two atoms with codes 6 and 39 have different central atoms so the result of the equation is zero. On the other hand the two atoms with codes 6 and 10 have the same central atoms but they have not similar edges so the result of the equation is one.

4. Atoms Similarity Kernel Function

4.1. Similarity Vector

Several extensions of Treelet kernel (Gaüzère *et al.*, 2012) were proposed by (Gaüzère *et al.*, 2015) which can be used to solve chemoinformatics problems. These extensions aim to weight each pattern according to its influence, to include the comparison of non-isomorphic patterns, to include stereo information and finally to explicitly encode cyclic information into kernel computation. They built their method on the hypothesis that similar molecules have similar properties. They also extended that hypothesis to consider that similar substructures may have a similar influence on molecular properties. Depending on that hypothesis they built kernel method that uses similarity between patterns to improve Treelet kernel.

The atoms common subgraphs matrix is the base from which the similarity between chemical compounds is computed. In the atoms similarity kernel function we use the atoms common subgraphs matrix to present a relationship between atoms stars. In this section we define similarity between two chemical compounds and how it is computed. Let we have chemical compound g_i then it has a corresponding list of atoms codes C_i , where C_i is defined by Equation (5).

$$C_i = \{(c, r) : c \text{ an atom code in } g_i, \text{ and } r \text{ frequency in } g_i\} \quad (5)$$

For any two chemical compounds g_i, g_j we define a union atoms codes list $C^U_{i,j}$ between their atoms codes lists C_i and C_j by Equation (6).

$$C^U_{i,j} = \left\{(c, r) : (c, r_i) \in C_i \text{ or } (c, r_j) \in C_j \text{ and } r = \max(r_i, r_j)\right\} \quad (6)$$

We denote by $g^s_{i,j}$ the similarity between two chemical compounds g_i and g_j and it is computed by finding the summation of the number of all subgraphs between each two star subgraphs in both C_i and C_j then that summation is divided by the summation of the number of all subgraphs between each two star subgraphs in $C^U_{i,j}$ and itself. The number of common subgraphs between each two atoms in two compounds is retrieved from the corresponding entity in the atoms common subgraphs matrix S. Equation (7), which computes $g^s_{i,j}$ is defined as the following:

$$g^s_{i,j} = \frac{\sum_{(x,r_x)\in C_i} \sum_{(y,r_y)\in C_j} r_x r_y S_{I(x)I(y)}}{\sum_{(w,r_w)\in C^U_{i,j}} \sum_{(z,r_z)\in C^U_{i,j}} r_w r_z S_{I(w)I(z)}} \quad (7)$$

Where x, y, w, and z are atom codes and $I(x), I(y), I(w)$, and $I(z)$ are the index of star subgraphs corresponding to x, y, w, and z respectively in the atoms common subgraphs matrix S. The number of common subgraphs between stars x, and y is the element $S_{I(x)I(y)}$ in matrix S. As the same, the number of common subgraphs between stars w, and z is the element $S_{I(w)I(z)}$ in matrix S.

It is clear that if two chemical compounds g_i and g_j are identical, C_i and C_j will be equal to each other and also the union atoms codes list $C^U_{i,j}$ will be equal to them. Because all atoms codes lists are equal to each other, the value of numerator will be equal to the value of denominator in Equation (7) and the similarity value will be $g^s_{i,j} = 1$. On the other hand, if there are no common atoms or subgraphs between g_i and g_j, the value of numerator in Equation (7) will be equal to zero and hence the similarity value will be $g^s_{i,j} = 0$. The more same atoms contained in g_i and g_j, the more same atoms codes contained in thier codes lists C_i and C_j and so $g^s_{i,j}$ will be more close to 1, vice versa the more different atoms lead to more different atoms codes and so $g^s_{i,j}$ will be more close to 0.

Similarity vector v^s_i for a given chemical compound g_i which belongs to a dataset D of chemical compounds is the vector that contains the similarity between g_i and each $g_j \in D$. Similarity vector v^s_i is defined by Equation (8).

$$v^s_i = \left(g^s_{i,1}, g^s_{i,2}, \cdots, g^s_{i,h}\right) \quad (8)$$

Where h is the size of the dataset D. The similarity vector v_i^s for a given chemical compound g_i will represent the set of features of g_i in the following sections. These sets of features (similarity vectors of molecules) will be used to create two kernel functions in the following sections.

4.2. Features Selection

Some of the set of features found within a dataset may be irrelevant to explain a given property. Such irrelevant features may induce useless calculus (increase training time) and decrease performances of classification algorithms. This problem is known as over-fitting which is related to classification specifically, and machine learning in general. Data over-fitting arises when the number of features is large and the number of training set is comparatively small. To overcome this problem we need to find ways to reduce the dimensionality of the feature space. In this paper three types of features selection (forward selection (FS), backward selection (BS), and ranked sequential forward selection (RSFS)) were studied. All these selection methods were applied on the features (similarity vectors of molecules) to select the subset of features that provides the best prediction accuracy. Forward selection and backward selection which were described in (Gaüzère *et al*., 2012) were used with using prediction accuracy instead of Residual Sum of Squares. The ranked sequential forward selection (RSFS) used to select features from similarity vectors is shown in Figure 4.

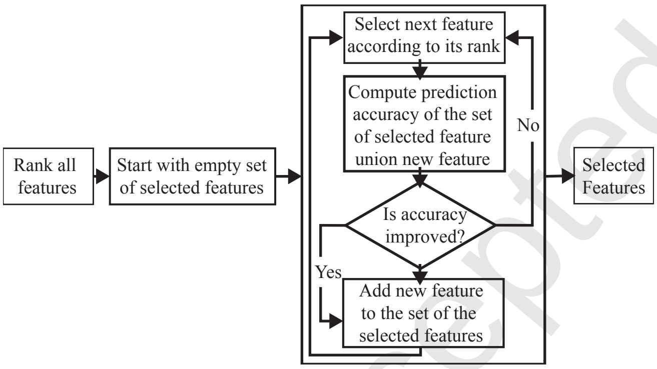

Rank all features
Start with empty set of selected features
Select next feature according to its rank
Compute prediction accuracy of the set of selected feature union new feature
Is accuracy improved?
Yes
No
Add new feature to the set of the selected features
Selected Features

Figure 4: The steps of sequential forward selection (RSFS).

In ranked sequential forward selection, we first rank features set according to the weight assigned to features by the SVM (Guyon *et al*., 2002). This ranking method is done by firstly using all features to train SVM then computing square of the weight assigned by the trained SVM to each feature and finally features are ranked according to these squared weights. According to that rank, features are sequentially added to an empty candidate set until the addition of further features does not improve the prediction. After all features had been tested, we had the selected set of features.

4.3. Kernel Function

We propose two kernel functions. The first one uses the similarity vectors as features for a polynomial kernel function of order two and we called it the atom similarity kernel K_s. The second kernel function uses the outputs of Equation (7) as kernel matrix entities and we called it the direct kernel K_d. The first kernel is define as the following. Let D is a dataset with h chemical compounds. The atom similarity kernel K_s between two compounds g_i and $g_j \in D$ is given by Equation (9), where similarity vectors of g_i and g_j are v_i^s and v_j^s.

$$K_s\left(g_i, g_j\right) = K\left(v_i^s, v_j^s\right) \tag{9}$$

The kernel function K between v_i^s and v_j^s corresponds to a definite positive kernel between vectors. In our experiments (Section 5), the RBF, scalar and polynomial kernels have been tested in order to select the best one and we find that the second degree polynomial kernel giving us the best results. The form of the kernel equation is given by Equation (10).

$$K(a, b) = \left(a^T \times b + 1\right)^2 \tag{10}$$

Where a and b are the similarity vectors. The second kernel function (direct kernel K_d) is given by Equation (11). In the K_d, the results of similarity between each two molecules is representing the kernel matrix of the direct kernel function K_d.

$$K_d\left(g_i, g_j\right) = g_{i,j}^s \tag{11}$$

The first kernel differs from the second kernel in how it treats the similarity vectors. In the first kernel the similarity of any two molecules do not depend only on the similarity between the structures of these two molecules but also it includes a similarity measure of how they both are similar to all the other molecules.

5. Experimental Results

5.1. Datasets

Two experimental evaluations for the proposed approach are done by using two chemical compounds datasets. The first one is the monoamine oxidase (MAO) dataset which is composed of 68 molecules and is divided into two classes: 38 molecules inhibit the monoamine oxidase (antidepressant drugs) and 30 do not. These chemical compounds are encoded as labeled graphs. The mean size of 68 molecules is 18.4 atoms for molecular. The mean degree of atoms in this dataset is 2.1 edges. The smallest molecular has 11 atoms and the largest one has 27 atoms. All databases in this section are available on the IAPR-TC15 Web page .

The second dataset is defined from the AIDS Antiviral Screen Database of Active Compounds and it is composed of 2000 chemical compounds, some of them are disconnected. These chemical compounds have been screened as active or inactive against HIV and they are splitted into three different sets. The first set is the train set which is composed of 250 compounds and is used to train SVM. The second one is the validation set which is composed of 250 compounds and is used to find parameters giving the best accuracy result. the third set is the test set which is composed of remaining 1500 compounds used to test the classification model (Riesen and Bunke, 2008).The mean size of molecules is 15.7 atoms for molecular. The mean degree of atoms in this dataset is 2.1 edges. The smallest molecular has 2 atoms and the largest one has 95 atoms. Prediction accuracy is the overall accuracy for the prediction which equals the division of the correct predictions by the all predictions. Prediction accuracy is given by Equation (12).

$$\text{Prediction Accuracy} = \frac{\text{Correct Predictions}}{\text{All Predictions}} \tag{12}$$

5.2. Comparative Results

Here, we present the results of implementing the proposed kernel functions on two datasets and compare the results with other methods. The first implementation is done on the monoamine oxidase (MAO) dataset. Classification accuracy is measured for each method using a leave one out procedure with a two-class SVM. This classification scheme is made for each of the 68 molecules of the dataset. Results of the different methods are presented in Table 1. Table 1 is divided into three parts. First part in the Table 1 shows the results of each method using a leave one out procedure with a two-class SVM. The second part contains the results of applying some selection methods on some methods (With SVM). the final part shows the result of using artificial neural network (ANN) on the matrix that consists of the similarity vectors.

Table 1: Prediction accuracy on the monoamine oxidase (MAO) dataset.

No.	Method	Prediction Accuracy
(1)	Suard et al. (2002)	80% (55/68)
(2)	Vishwanathan et al. (2010)	82% (56/68)
(3)	Neuhaus and Bunke (2007)	90% (61/68)
(4)	Riesen et al. (2007)	91% (62/68)
(5)	Normalized standard Graph Laplacian kernel (2012)	90% (61/68)
(6)	Normalized fast Graph Laplacian kernel(2012)	90% (61/68)
(7)	Mahé and Vert (2008)	96% (65/68)
(8)	Gaüzère et al. (2012)	94% (64/68)
(9)	Atoms similarity kernel function (K_s)	90% (61/68)
(10)	Direct kernel function	91% (62/68)
(11)	(Treelet + MKL) Gaüzère et al. (2013)	96% (65/68)
(12)	K_s (with RSFS)	98.5% (67/68)
(13)	K_s (with FS)	96% (65/68)
(14)	K_s (with BS)	97% (66/68)
(15)	Similarity Vectors + ANN	98.5% (67/68)

The first two methods in Table 1 are based on linear patterns. The first one (Suard *et al.*, 2002) computes the similarity between two graphs from the average similarity between each pair of paths which are extracted from both graphs and has accuracy equals to 80%. The second method (Vishwanathan *et al.*, 2010) counts the number of identical random walks of two graphs and with accuracy equals to 82%. The third one (Neuhaus and Bunke, 2007) is a Gaussian kernel on the suboptimal edit distance which has accuracy equals to 90% and the fourth method (Riesen *et al.*, 2007) corresponds to an embedding of the graphs into a vector which encodes the edit distance between the graphs and a set of selected prototypes and has accuracy equals to 91%.

Lines 5 and 6 correspond to graph Laplacian kernel (Gaüzère *et al.*, 2012) which uses a suboptimal graph edit distance with accuracy equals to 90%. Then, the next two lines correspond to methods which are based on non linear patterns. The seventh method (Mahé and Vert, 2008) computes the similarity between two graphs from the number of common tree-patterns and treelet kernel (Gaüzère *et al.*, 2012) which is based on subtrees enumeration with accuracy equals 96% and 94% respectively.The last two methods, in first part of the table, correspond to the proposed kernel functions which are defined in section 4. The first one of them corresponds to passing all features vector (similarity vector) to atoms similarity kernel function (K_s) with accuracy equals to 90%. The second one corresponds to using the direct kernel function (K_d) and has accuracy equals to 91%.

The second part of the table corresponds to using the ranked sequential forward selection (RSFS), Forward selection (FS), and Backward selection (BS) methods on atoms similarity kernel function (K_s) which is discussed in section 4.2. Line 11 corresponds to selection of relevant treelets via multiple kernel learning (MKL) (Gaüzère *et al.*, 2013) with accuracy equals to 96%. The RSFS selection method with (K_s) (line 12) determined that, after ranking all features, the best features are the first five features with prediction accuracy equals to 98.5%. Line 13 corresponds to (K_s) with forward selection which selects 5 features and has accuracy equals to 96%. Line 14 corresponds to (K_s) with backward selection which selects 18 features and has accuracy equals to 97%. The last line corresponds to using ANN on the matrix consists of all similarity vectors and it has prediction accuracy 98.5%.

In our method we encode only 24 different atoms, from them we construct similarity vectors as we described in section 4. Features selection have improved the accuracy of the prediction and by using only five elements of similarity vector we can reduce the time needed for training step.

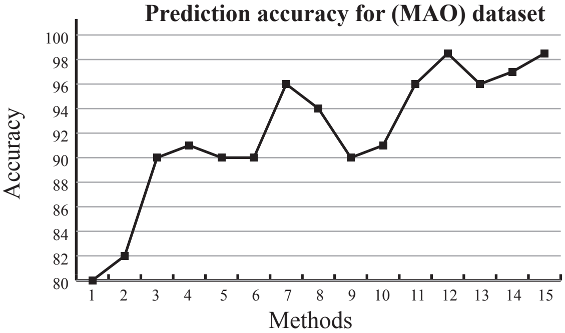


Figure 5: The prediction accuracy of all methods on MAO dataset.

Figure 5 shows that the two methods in line 12 and line 15, atoms similarity kernel with RSFS features selection and the (similarity vectors + ANN), have the best prediction accuracy which equals to 98.5%. The next one is atoms similarity kernel with BS features selection with prediction accuracy equals to 97%. The next best three methods are the seventh method, treelet + MKL, and atoms similarity kernel with FS features selection with accuracy equals 96%. The next one is the treelet with accuracy equals 94%. The next two are the fourth method and direct kernel method with accuracy equals 91%. This method is followed by the third method and both types of normalized graph laplacian kernel where all of them have accuracy equals 90%. Moreover, the atoms similarity kernel without features selection has 90% prediction accuracy. Then there are two reaming methods, the second and first one shown in Table 1 have accuracy equals to 82% and 80% respectively.

It is clear that both K_s (the atoms similarity kernel) with 90% and K_d (direct kernel) with 91% accuracy have competitive prediction accuracy with all tested methods. It is also clear that atoms similarity kernel with RSFS features selection has the best prediction accuracy equals to 98.5% among all methods that use features selection.

Table 2: Prediction accuracy on the AIDS antiviral screen database of active compounds.

No.	Method	Prediction Accuracy
(1)	Riesen and Bunke (2008)	97.3%
(2)	Suard et al. (2002)	98.5%
(3)	Vishwanathan et al. (2010)	98.5%
(4)	Neuhaus and Bunke (2007)	99.7%
(5)	Riesen et al. (2007)	98.2%
(6)	Laplacian kernel (2012)	99.3%
(7)	Gaüzère et al. (2012)	99.1%
(8)	Atoms similarity kernel function (K_s)	99.2%
(9)	Direct kernel function	91.9%
(10)	(Treelet+MKL) Gaüzère et al. (2013)	99.7%
(11)	K_s (with RSFS)	99.2%
(12)	K_s (with FS)	98.9%
(13)	K_s (with BS)	99.4%
(14)	Similarity Vectors + ANN	99.4%

The second implementation is done on AIDS dataset with SVM and the results of the different methods are presented in Table 2. This table is also divided into three parts like Table 1. First part in the Table 2 shows the results of each method using a leave one out procedure with a two-class SVM. The second part contains the results of applying some selection methods on some methods (With SVM). The final part shows the result of using artificial neural network (ANN) on the matrix that consists of the similarity vectors.

The first method in Table 2 is based on using the graph edit distance and k-Nearest Neighbor classifier and has accuracy equals to 97.3% (Riesen and Bunke, 2008). The second method (Suard *et al.*, 2002) computes the similarity between two graphs by using paths extracted from both graphs and has accuracy equals to 98.5%. The third method (Vishwanathan *et al.*, 2010) is based on identical random walks of two graphs and has accuracy equals to 98.5%. The fourth method (Neuhaus and Bunke, 2007) is a Gaussian kernel on the suboptimal edit distance and has accuracy equals 99.7%. The fifth method (Riesen *et al.*, 2007) is based on an embedding of the graphs into a vector which encodes the edit distance between the graphs and a set of selected prototypes and has accuracy equals to 98.2%.Line 6 corresponds to graph Laplacian kernel (Gaüzère *et al.*, 2012) using a suboptimal graph edit distance with accuracy equals to 99.3%. The seventh method (Gaüzère *et al.*, 2012) is based on subtrees enumeration with accuracy equals 99.1%.Lines 8 and 9 correspond to the proposed kernel functions in section 4. The first one of them corresponds to passing all features vector (similarity vector) to atoms similarity kernel function (K_s) with accuracy equals to 99.2%. The second one corresponds to using the direct kernel function (K_d) and has accuracy equals to 91.9%.

The second part of the table corresponds to applying the ranked sequential forward selection (RSFS), Forward selection (FS), and Backward selection (BS) methods on atoms similarity kernel function (K_s) which is discussed in section 4.2. Line 10 corresponds to selection of relevant treelets via multiple kernel learning (MKL) (Gaüzère *et al.*, 2013) with accuracy equals to 99.7%. The RSFS selection method with (K_s) (line 11) determined that, after ranking all features, the best features are the first 162 features with prediction accuracy equals to 99.2%. Line 12 corresponds to (K_s) with forward selection which selects 7 features and has accuracy equals to 98.9%. Line 13 corresponds to (K_s) with backward selection which selects 248 features and has accuracy equals to 99.4%. The last line corresponds to applying ANN on the matrix which consists of all similarity vectors and it has prediction accuracy 99.4%. In our method we encode only 486 different atoms, from them we construct similarity vectors as we described in section 4.

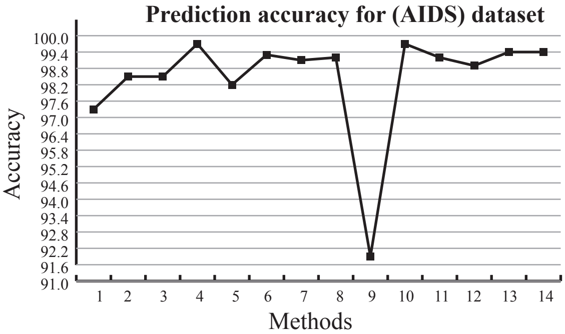


Figure 6: The prediction accuracy of all methods on AIDS dataset.

Figure 6 shows that two methods, the fourth method and Treelet+MKL method, have the best prediction accuracy which equals to 99.7%. The next two methods are the atoms similarity kernel with BS features selection and the (similarity vectors + ANN) with accuracy equals 99.4%. The next best method is the Laplacian kernel method with accuracy equals 99.3%. The next two are the atoms similarity kernel method with and without RSFS features selection step with accuracy equals 99.2%. Then the seventh method, treelet kernel with accuracy equals 99.1%. The next one is the atoms similarity kernel with FS features selection with accuracy equals 98.9%. Then there are two methods, the second one and the third one which have the same accuracy 98.5%. The next two methods are the fifth method and the first method with accuracy equal to 98.2% and 97.3% respectively. The last one is direct kernel with accuracy 91.9%.

It is clear that the atoms similarity kernel with 99.2% accuracy has competitive prediction accuracy with all tested methods. Also, it is clear that atoms similarity kernel with BS features selection has the second best prediction accuracy equals to 99.4% among all methods that use features selection. Moreover, the atoms similarity kernel function is more stable than direct kernel function in both two datasets. Atoms similarity kernel function derives its strength from its way of computation. Atoms similarity kernel function is more stable because it computes the similarity of two molecules depending on the similarity between their structures, and combines it with the similarity of these two molecules' structures with the other molecules in the dataset.

5.3. Complexity of the Approach

For computing complexity of our approach, let we have a dataset D which contains h chemical compounds and the largest compound in that dataset contains p of atoms and the maximum number of atom neighbors is t (atom degree). There are three steps to create the similarity vector. First step is to code all atoms in dataset and to do that we create binary tree as data structure to store all atoms codes. The node of the binary tree holds the atom code, also it holds the order of that code among all atoms codes in an ascending order. Also a pointer of that node is stored in all atoms which have the same code and by this way retrieving any data related to that code will cost constant time. Number of unique atoms codes will approximately equals to p and so the coding of all atom codes for all dataset will be of $O(htp)$, where computing code needs to scan all atom neighbors. Sorting atoms code by using balanced binary tree will cost $O(p \log p)$.

The second step is to create the atoms common subgraphs matrix S. To create S we need to apply Equation (4) for each one of the atoms codes. Worst case for Equation (4) is $O(t^2)$ and will happen if the number of common primes between two atoms codes n is equal to the maximum degree t of the atom. Because the number of unique atoms codes is p, the construction of the atoms common subgraphs matrix S will cost $O(p^2t^2)$.

The final step is to construct similarity vector v_i^s for each compound g_i. Construction of v_i^s needs to compute the Equation (7) h times so the all complexity equals $O(hp^2)$ for one compound and equals $O(h^2p^2)$ for all compounds in dataset. As h is greater than t so $O(h^2p^2)$ is greater than both $O(htp)$ and $O(p^2t^2)$. From the complexity of the previous three steps, it is clear that overall complexity equals $O(h^2p^2)$, where the number of compounds is h and the largest compound in that dataset contains p of atoms.

Table 3: Running time in second to create kernel matrix for proposed kernel function.

Dataset	Kernel Size	Time
MAO	68 by 68 matrix	0.0625 s
AIDS for kernel matrix only	250 by 250 matrix	1.5991 s
AIDS for all similarity vectors	250 by 1750 matrix	12.793 s

The computation times required to compute the kernel matrices of the proposed kernel function is shown in Table 3. These computation times are done in PC with (Pentium(R) Dual-Core CPU E5300 2.60 GHz and 2.75 GB of RAM). These times only represent the times required to create kernel matrix only without selection or prediction. Moreover time required to read input files is not included.

6. Conclusions and Future Works

In this paper we have proposed two new kernel functions to solve activity prediction problems. Two kernel functions are based on atoms similarity which is computed by counting common subgraphs between neighbors of each two atoms. To reduce the time which is needed to find the relationship between atoms we present a coding system which maps each atom to a unique number. This number can be used to find common subgraph between atoms.

The two kernel functions are applied on MAO, a chemical dataset with 68 chemical compounds. The results of experiments show that atom similarity kernel function has prediction accuracy equals to 90% and by ranking and applying sequential forward selection for feature vector (similarity vector) the prediction accuracy is improved to 98.5%. Hence, it is clear that features selection reduces data over-fitting problem and improve prediction accuracy. Also, results shows that direct kernel function has prediction accuracy equals to 91% then the two kernel functions are also applied on AIDS dataset with 2000 chemical compounds. The results of experiments shows that atom similarity kernel function has prediction accuracy equals to 99.2% and by ranking and applying sequential forward selection for feature vector (similarity vector) the prediction accuracy also equals 99.2%. Also, backward selection has prediction accuracy equals to 99.4%. Experimental results show that the proposed approach has an advanced place in the accuracy comparison and so the proposed approach has a competitive prediction accuracy with all tested methods.

The proposed approach can be modified to work on finding similarity between two chemical compounds by using similarity between bigger subgraphs like edges or paths and so on. Tests on other datasets for predicting the biological activity of molecule (drug properties, toxicity and regression problems) can be carried on. Also, there is a challenge in reducing complexity of constructing similarity vectors and new functions can be used to improve computation time.

Abu El-Atta, A. H., Moussa, M. I., and Hassanien, A. E. (2014) Predicting biological activity of 2, 4, 6-trisubstituted 1, 3, 5-triazines using random forest, *Proc. of the 5th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2014)*, 101-110.

Barbu, E., Raveaux, R., Locteau, H., Adam, S., and Heroux, P. (2006) Graph classification using genetic algorithm and graph probing application to symbol recognition, *Proc. 18th Intl Conf. Pattern Recognition (ICPR)*.

Brun, L., Conte, D., Foggia, P., Vento, M., and Villemin, D. (2010) Symbolic learning vs. graph kernels: An experimental comparison in a chemical application, *In: Proceedings of the 14th Conference on Advances in Databases and Information Systems (ADBIS 2010)*, 31-40.

Cherqaoui, D., and Villemin, D. (1994) Use of a neural network to determine the boiling point of alkanes, *J. Chem. Soc. Faraday Trans.*, **90**,97-102.

Deshpande, M., Kuramochi, M., and Karypis, G. (2005) Frequent substructure-based approaches for classifying chemical compounds, *IEEE Trans. Knowledge and Data Eng.*, **17(8)**, 1036-1050.

Fröhlich, H., Wegner, J., Sieker, F. and Zell, A. (2006) Kernel functions for attributed molecular graphs-A new similarity-based approach to ADME prediction in classification and regression, *QSAR & Combinatorial Science*, **25**, 317-326.

Gaüzère, B. (2013) Graph kernels for the prediction of molecular properties, *Cheminformatics, PhD Thesis, University of Caen*.

Gaüzère, B., Brun, L., and Villemin, D. (2012) Two new graphs kernels in chemoinformatics, *Pattern Recognition Letters*, **33(15)**, 2038-2047.

Gaüzère, B., Grenier, P., Brun, L., and Villemin, D. (2015) Treelet kernel incorporating cyclic, stereo and inter pattern information in Chemoinformatics, *Pattern Recognition*, **48(2)**, 356-367.

Grenier, P., Brun, L., and Villemin, D. (2014) Incorporating molecules stereisomerism within the machine learning framework, *Structural, Syntactic, and Statistical Pattern Recognition Lecture Notes in Computer Science*, **8621**, 12-21.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002) Gene selection for cancer classification using support vector machines, *Machine Learning*, **46**, 389-422.

Horvath, T., Gartner, T., and Wrobel, S. (2004) Cyclic pattern kernels for predictive graph mining, *Proc. ACM SIGKDD*.

Huan, J., Wang, W., Washington, A., Prins, J., Shah, R., and Tropsha, A. (2004) Accurate classification of protein structural families based on coherent subgraph analysis, *Proc. Pacific Symp. Biocomputing (PSB)*, 411-422.

Huang, Y., Li, H., Hu, H., Yan, X., Waterman, M.S., Huang, H., and Zhou, X.J. (2007) Systematic discovery of functional modules and context-specific functional annotation of human genome, *Bioinformatics*, **23**, 222-229.

Jin, R., Mccalle, S., and Almaas, E. (2007) Trend Motif: a graph mining approach for analysis of dynamic complex networks, *Proc. IEEE Intl Conf. Data Mining (ICDM)*, 541-546.

Kashima, H., Tsuda, K., and Inokuchi, A. (2003) Marginalized kernels between labeled graphs, *Proc. 20th Intl Conf. Machine Learning (ICML)*.

Mahé, P., and Vert, J. P. (2008) Graph kernels based on tree patterns for molecules, *Machine Learning*, **75(1)**, 3-35.

Morgan, H. L. (1965) The generation of a unique machine description for chemical structuresa technique developed at chemical abstracts service, *Journal of Chemical Documentation*, **5(2)**, 107-113.

Neuhaus, M., and Bunke, H. (2007) Bridging the gap between graph edit distance and kernel machines, *World Scientific Pub Co Inc.*.

Poezevara, G., Cuissart, B., and Crémilleux, B. (2009) Discovering emerging graph patterns from chemicals, *In Proc. 18th Internat. Symp. on Methodologies for Intelligent Systems (ISMIS 2009), LNCS, Prague*, 45-55.

Ralaivola, L., Swamidass, S.J., and Saigo, H. (2005) Graph kernels for chemical informatics, *Neural Networks*, **18**, 1093-1110.

Riesen, K. and Bunke, H. (2008) IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning, *da Vitora Lobo, N. et al. (Eds.), SSPR&SPR 2008, LNCS*, **5342**, 287-297.

Riesen, K., Neuhaus, M., and Bunke, H. (2007) Graph embedding in vector spaces by means of prototype selection, *In: Escolano, F., and Vento, M. (Eds.), 6th IAPR-TC15 Internat. Workshop GbRPR 2007. IAPR-TC15. Springer-Verlag*, 383-393.

Smalter, A., Huan, J., Jia, Y., and Lushington, G. (2010) GPD: a graph pattern diffusion kernel for accurate graph classification with applications in cheminformatics, *IEEE/ACM Trans Comput Biol Bioinform*, **7(2)**, 197-207.

Smalter, A., Huan, J., and Lushington, G. (2008) Structure-based pattern mining for chemical compound classification, *Proc. Sixth Asia Pacific Bioinformatics Conf.*.

Suard, F., Rakotomamonjy, A., and Bensrhair, A. (2002) Kernel on bag of paths for measuring similarity of shapes, *In Proc. European Symposium on Artificial Neural Networks (ESANN)*, 355-360.

Todeschini, R., Consonni, V. (2009) Molecular descriptors for chemoinformatics (2 volumes), *Wiley-VCH Verlag GmbH, Weinheim, Germany*, **V(1)**, 395-397.

Turner, J. V., Maddalena, D. J., and Cutler, D. J. (2004) Pharmacokinetic parameter prediction from drug structure using artificial neural networks, *Int. J. Pharm.*, **270**, 209-219.

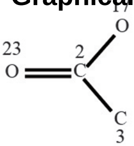
Vishwanathan, S., Borgwardt, K. M., Kondor, I. R., and Schraudolph, N. N. (2010) Graph kernels, *Res.*, **11**, 1201-1242.

Weininger, D., Weininger, A., and Weininger, J. L. (1989) SMILES. 2. Algorithm for generation of unique SMILES notation, *Journal of Chemical Information and Computer Sciences*, **29(2)**, 97-101.

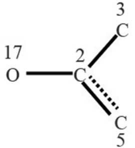
Weston, J., Kuang, R., Leslie, C., and Noble, W. S. (2006) Protein ranking by semi-supervised network propagation, *BMC Bioinformatics*, **7**(Suppl 1):S10. doi: 10.1186/1471-2105-7-S1-S10.

Ahmed H. Abu El-Atta, M. I. Moussa, Aboul Ella Hassanien: Predicting Biological Activity of 2, 4, 6-trisubstituted 1, 3, 5-triazines Using Random Forest. IBICA 2014: 101-110

Graphical Abstract



	Carbon Atom	Oxygen Atom
Atom Type	**2**	**13**
Single Bond	**3**	**17**
Aromatic Bond	**5**	**19**
Double Bond	**7**	**23**
Triple Bond	**11**	**29**



- We proposed new kernel functions to predict activity of molecule.
- Atoms coding system based on prime numbers is proposed.
- The proposed functions were tested on two datasets and competitive results were obtained.