Current Topics in Medicinal Chemistry, 2020, *20*, 1582-1592

RESEARCH ARTICLE

Learning the Edit Costs of Graph Edit Distance Applied to Ligand-Based Virtual Screening

Carlos Garcia-Hernandez[1], Alberto Fernández[1,*] and Francesc Serratosa[2]

[1]*Department of Chemical Engineering, Rovira i Virgili University, Tarragona, Spain;* [2]*Department of Computer Engineering and Mathematics, Rovira i Virgili University, Tarragona, Spain*

ARTICLE HISTORY

Received: October 23, 2019
Revised: November 19, 2019
Accepted: December 07, 2019

DOI:
10.2174/1568026620666200603122000

Abstract: ***Background*:** Graph edit distance is a methodology used to solve error-tolerant graph matching. This methodology estimates a distance between two graphs by determining the minimum number of modifications required to transform one graph into the other. These modifications, known as edit operations, have an edit cost associated that has to be determined depending on the problem.

***Objective*:** This study focuses on the use of optimization techniques in order to learn the edit costs used when comparing graphs by means of the graph edit distance.

***Methods*:** Graphs represent reduced structural representations of molecules using pharmacophore-type node descriptions to encode the relevant molecular properties. This reduction technique is known as extended reduced graphs. The screening and statistical tools available on the ligand-based virtual screening benchmarking platform and the RDKit were used.

***Results*:** In the experiments, the graph edit distance using learned costs performed better or equally good than using predefined costs. This is exemplified with six publicly available datasets: DUD-E, MUV, GLL&GDD, CAPST, NRLiSt BDB, and ULS-UDS.

***Conclusion*:** This study shows that the graph edit distance along with learned edit costs is useful to identify bioactivity similarities in a structurally diverse group of molecules. Furthermore, the target-specific edit costs might provide useful structure-activity information for future drug-design efforts.

Keywords: Structure-activity relationships, Graph edit distance, Extended reduced graph, Virtual screening, Molecular similarity, Machine learning.

1. INTRODUCTION

Quantitative structure-activity relationship (QSAR) models are computational or mathematical models that attempt to find a significant correlation between molecular structure and molecular activity. With the huge increment in the amount of available data about chemical compounds and their reactivity, there is as well, a rising need for computational tools to reduce the drug synthesis and test cycle execution times. These tools are essential if activity data are to be analyzed and new models created for virtual screening techniques [1].

Virtual screening - usually referred to as the use of computational techniques to search and filter chemical databases [2, 3] - is a common step in the drug discovery process. Two main categories of methods can be found in the virtual screening inventory: structure-based virtual screening (SBVS) [4] and ligand-based virtual screening (LBVS) [5]. SBVS uses the 3D structure information of a target (obtained from X-ray, NMR, or some other method), to dock a group of molecules into the binding site of a protein, and estimate the likelihood that the molecules will bind to the protein [6, 7]. LBVS uses information about the known activity of some molecules, activity in terms of their behavior as ligands that binds to a receptor, to predict the unknown activity of new molecules [8]. In this work, the focus will be only on LBVS applications. The main LBVS approaches are pharmacophore mapping [8], shape-based similarity [9], fingerprint similarity, and various machine learning methods [10]. The measure of molecular similarity used is an important feature in the context of LBVS, which can determine the degree of success of a virtual screening method.

It is assumed that structurally similar molecules are likely to have similar activity properties [11]; therefore, molecular similarity methods are commonly used to select good candidates in the drug discovery industry. These similarity methods are used in applications related to molecular clustering, similarity searching or molecular screening [12-16].

Regardless of the application, molecular similarity searching usually requires one descriptor representing the molecules and a distance measure by which the level of similarity (or dissimilarity) between those molecules can be numerically defined. Different types of descriptors have been used [3, 17, 18], which are frequently classified as one-

*Address correspondence to this author at the Department of Chemical Engineering, Rovira i Virgili University, Tarragona, Spain;
E-mail: alberto.fernandez@urv.cat

1873-4294/20 $65.00+.00

dimensional (1D), two-dimensional (2D) or three-dimensional (3D) depending on the molecular in- formation used to compute them [19]. 1D descriptors include general molecular properties such as size, molecular weight, logP, dipole moment or BCUT parameters [20-23]. 2D descriptors create array representations of the molecules by simplifying the atomic information within them such as 2D fingerprints [24-27]. 3D descriptors use the 3D information such as molecular volume [28, 29]. Additionally, there are methods representing compounds as trees [30], or graphs [31, 32]. Among the graph options, some of these methods represent the compounds using reduced graphs [33-36], which group atomic sub-structures together in terms of related features such as pharmacophoric features, hydrogen-bonding, ring systems or other rules. Likewise, extended reduced graphs (ErG) [36] is an extension of the reduced graphs described by Gillet *et al.* [35], which introduces some changes in order to better represent the size, shape, and pharmacophoric properties of the molecules.

ErGs have demonstrated to be a powerful tool for virtual screening [36], it can be used as an abstraction layer from the complex physico-atomic world, and leave the path clear to work directly with the pharmacophoric chemical information inside the molecular structures.

Three similarity measures have been used to perform reduced graph comparisons. The first one maps the reduced graphs into a 2D fingerprint [35-37], the second one maps them into sets of shortest paths [38], and the third one makes the comparison directly on the graphs *via* the graph edit distance method [39].

The main goal of this study is to implement optimization tools in order to improve the recognition ratio when classifying molecules represented as ErGs according to their biological activity. The optimization techniques help to determine better values to be considered as edit costs in the process of computing the graph edit distance (GED) [40-43] used as dissimilarity measure between molecular graphs based on ErGs. GED considers the distance between two graphs as the minimum cost of modifications required to transform one graph into another. Each modification can be one of the following six operations: insertion, deletion and substitution of both nodes and edges in the graph.

In a previous work [39], we used the edit costs proposed by Harper *et al.* [38], which were assigned by experts considering the different node and edge types. In this paper, we present a method for optimizing those edit costs, based on minimizing the distance between molecules correctly classified and maximizing the distance between molecules incorrectly classified.

The process of mathematical optimization is the selection of the best element into a set of alternatives, taking into account some criteria or constraints. This optimization process minimizes or maximizes an objective function, by iteratively using different input values and computing the output.

Optimization techniques are used in several areas like mechanics, finance, engineering, physics, biology, molecular modeling, etc. Particularly in molecular modeling applications, optimization tools are used along with QSAR descriptors to optimize the existing leads by structural modifications in order to improve a specific activity and mitigate or remove any possible side effects [44]. In the drug discovery process, a lead compound is a molecule having biological activity likely to be useful, but may require modifications to better fit the target. Having optimized QSAR models developed on the basis of the lead series can assist in optimizing the lead compounds and help them to overcome their drawbacks [45].

The process of optimization, taking into account target-specific structure-activity models working with activity-known hits, can be helpful for high-throughput screening applications by rapidly searching through the library and identifying the most promising molecular candidates. This kind of optimized screening can lower the number of high-throughput experiments giving the opportunity to perform more complex low-throughput ones [44]. Additionally, the optimization of the models gives a better insight into the chemical space regarding those compounds in proximity to the ligands.

This work is inspired in a similar one carried out by Birchall *et al.* [46], in which the authors optimize the transformation costs of a String Edit Distance-based method to compare molecules using reduced graphs. In contrast, our work optimizes the edit costs of a Graph Edit Distance-based method to compare molecules using ErG.

The outline of this paper is as follows. First, materials and methods are presented and explained in detail including the datasets used, the GED methodology and the optimization process. Second, we present the computational results. And third, the paper is concluded with a final discussion.

2. MATERIALS AND METHODS

2.1. Datasets

Six publicly available datasets were used in this study. They are: ULS and UDS [47], GDD and GLL [48], DUD Release 2 (DUD-E) [49], NRLiSt BDB [50], MUV [51], and a dataset from Comparative Analysis of Pharmacophore Screening Tools (CAPST) [52]. All these datasets were formatted and standardized in an easy-to-use form by the LBVS benchmarking platform developed by Skoda and Hoksza [53]. The concept and functionality of this platform are similar to that developed by Riniker and Landrum [54] including some extended features. The datasets within this platform consist of several selections of active and inactive molecules arranged according to the tar- get they were physically tested with. Each selection is separated into two sub-groups named test and train sets so that it can be readily used for machine learning applications. Table **1** shows all targets available in the datasets.

A subset of the first 100 active molecules and 100 inactive molecules were selected per target from the datasets as they were formatted in the LBVS benchmarking platform. In some cases, available active molecules are less than 100; for those cases, all available active molecules and the same number of inactive molecules are used. Then, we split the sets by half in order to have train and test subsets that were used independently, the former to optimize the transformation costs and the latter to evaluate the recognition ratio with unknown data.

Table 1. Input data used for the experiments. The column entitled 'Dataset' contains the name of each dataset, and the column entitled 'Targets used' contains the name of the targets used during the experiments for each dataset. Note that in the result plots shown below, per-target points are arranged in the same order as they are in this table.

Dataset	Targets Used
ULS-UDS	5HT1F_Agonist, MTR1B_Agonist, OPRM_Agonist, PE2R3_Antagonist
GLL&GDD	5HT1A_Agonist, 5HT1A_Antagonist, 5HT1D_Agonist, 5HT1D_Antagonist, 5HT1F_Agonist, 5HT2A_Antagonist, 5HT2B_Antagonist, 5HT2C_Agonist, 5HT2C_Antagonist, 5HT4R_Agonist, 5HT4R_Antagonist, AA1R_Agonist, AA1R_Antagonist, AA2AR_Antagonist, AA2BR_Antagonist, ACM1_Agonist, ACM2_Antagonist, ACM3_Antagonist, ADA1A_Antagonist, ADA1B_Antagonist, ADA1D_Antagonist, ADA2A_Agonist, ADA2A_Antagonist, ADA2B_Agonist, ADA2B_Antagonist, ADA2C_Agonist, ADA2C_Antagonist, ADRB1_Agonist, ADRB1_Antagonist, ADRB2_Agonist, ADRB2_Antagonist, ADRB3_Agonist, ADRB3_Antagonist, AG2R_Antagonist, BKRB1_Antagonist, BKRB2_Antagonist, CCKAR_Antagonist, CLTR1_Antagonist, DRD1_Antagonist, DRD2_Agonist, DRD2_Antagonist, DRD3_Antagonist, DRD4_Antagonist, EDNRA_Antagonist, EDNRB_Antagonist, GASR_Antagonist, HRH2_Antagonist, HRH3_Antagonist, LSHR_Antagonist, LT4R1_Antagonist, LT4R2_Antagonist,MTR1A_Agonist, MTR1B_Agonist, MTR1L_Agonist, NK1R_Antagonist, NK2R_Antagonist, NK3R_Antagonist, OPRD_Agonist, OPRK_Agonist, OPRM_Agonist, OXYR_Antagonist, PE2R1_Antagonist, PE2R2_Antagonist, PE2R3_Antagonist, PE2R4_Antagonist, TA2R_Antagonist, V1AR_Antagonist, V1BR_Antagonist, V2R_Antagonist
CAPST	CDK2, CHK1, PTP1B, UROKINASE
DUD-E	COX2, DHFR, EGFR, FGFR1, FXA, P38, PDGFRB, SRC, AA2AR
NRLiSt_BDB	AR_Agonist, AR_Antagonist, ER_Alpha_Agonist, ER_Alpha_Antagonist, ER_Beta_Agonist, FXR_Alpha_Agonist, GR_Agonist, GR_Antagonist, LXR_Alpha_Agonist, LXR_Beta_Agonist, MR_Antagonist, PPAR_Alpha_Agonist, PPAR_Beta_Agonist, PPAR_Gamma_Agonist, PR_Agonist, PR_Antagonist, PXR_Agonist, RAR_Alpha_Agonist, RAR_Beta_Agonist, RAR_Gamma_Agonist, RXR_Alpha_Agonist, RXR_Alpha_Antagonist, RXR_Gamma_Agonist, VDR_Agonist
MUV	466, 548, 600, 644, 652, 689, 692, 712, 713, 733, 737, 810, 832, 846, 852, 858, 859

2.2. Molecular Representation

Reduced graphs are smaller representations of the original atomic graph from a chemical compound, in which the main information is condensed in feature nodes to give summary abstractions of the chemical structures. Different versions of reduced graphs can be used [32, 34, 36-38], and they depend on the features they summarize or the use that is given to them. In the virtual screening context, the structures are reduced to track down features or substructures having the potential to interact with a specific receptor and, at the same time, trying to keep the topology and spatial distribution of those features.

The reduction methodology used in this study is the ErG described by Stiefl *et al.* [36], where node features represent pharmacophore-type node descriptions. As the authors point out, this methodology can be described as a hybrid approach between reduced graphs [35] and binding property pairs [55].

In ErGs, nodes can be a single or a combination of the following features: hydrogen-bond donor, hydrogen-bond acceptor, positive charge, negative charge, hydrophobic group and aromatic ring system. There are also featureless nodes, which work as links between the main features and help them to keep the spatial distribution among them. These featureless nodes can be carbon or non-carbon link nodes. Fig. (**1**) exhibits an example of an ErG. The upper half of the image shows a chemical substance with its pharmacophoric substructures highlighted, and the lower half shows the ErG obtained from that molecule.

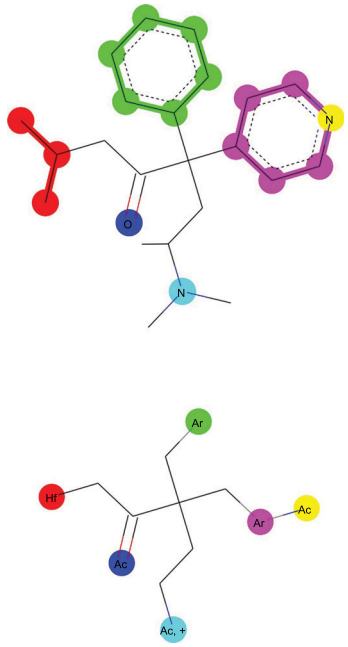

Fig. (1). Example of molecule reduction using ErG. The original molecule is at the top and its ErG representation is at the bottom. Ac: H-bond acceptor; Hf: hydrophobic group; Ar: aromatic ring system; +: positive charge. Colors are used to show how different parts of the original structure are reduced to nodes in the ErG.

2.3. Molecular Comparison

Once the molecules have been represented as ErGs, we compare them by means of the GED. Fig. (**2**) shows a molecular comparison procedure using the GED-based similarity method. The GED is defined as the minimum cost of modifications required to transform one graph into the other. These modifications are called edit operations and six of them have been used: insertion, deletion and substitution of both nodes and edges. For each pair of graphs A and B, there are several $editPath(A, B) = (\sigma_1, ..., \sigma_k)$ that transform one graph into the other, considering that each σ_i indicates an edit operation. An edit path holding the transformations from graph A into graph B can be seen in Fig. (**3**). In this case, the edit path consists of the following five edit operations: delete edge, delete node, insert node, insert edge, and substitute node. The substitution operation in the last step is needed since it is assumed that the attributes in both nodes are different.

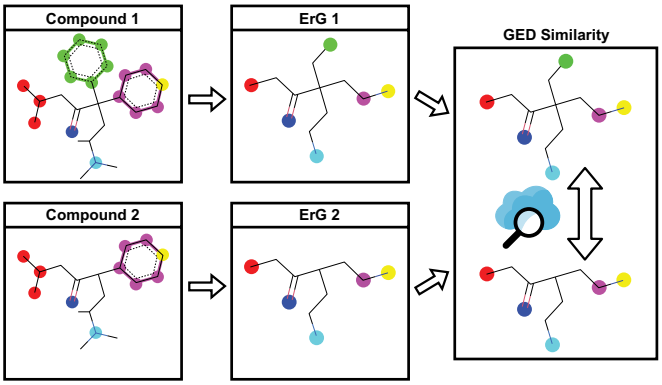

Fig. (2). Comparison of two molecules comprising two steps. First, we extract the ErGs; second, we apply the GED.

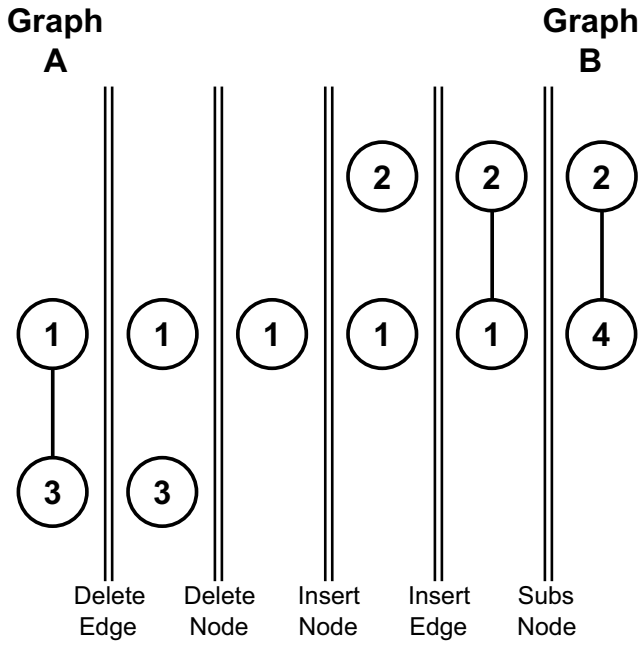

Fig. (3). An edit path that transforms graph A into graph B.

Edit costs have been introduced to quantitatively evaluate each edit operation and ultimately determine which edit path has the minimum total cost. The aim of the edit costs is to designate a coherent transformation penalty in proportion to the extent to which it modifies the transformation sequence. For instance, when ErGs are compared, it makes sense that the cost of substituting a "hydrogen-bond donor" feature with a joint "hydrogen-bond donor-acceptor" feature be less heavily penalized than the cost of substituting a "hydrogen-bond donor" feature with an "aromatic ring" system. Similarly, inserting a single bond should have a lower penalization cost than inserting a double bond, and so on.

In a previous work [39], we used the edit costs proposed by Harper *et al.* [38] with small changes to fit the ErG features. The node and edge descriptions are shown in Table **2** and the specific costs proposed by Harper *et al.* [38] are exposed in Tables **3** and **4**. Note that the insertion and deletion costs applied to a given node are constant. Moreover, substitutions are symmetric, which means that substitution of node type A to B is assigned the same cost as substitution of type B to A, to guarantee the symmetry property for the GED.

Table 2. Description of the node and edge attributes that compose an ErG.

Node Attributes	
Attribute	**Description**
[0]	Hydrogen-bond donor
[1]	Hydrogen-bond acceptor
[2]	Positive charge
[3]	Negative charge
[4]	Hydrophobic group
[5]	Aromatic ring system
[6]	Carbon link node
[7]	Non-carbon link node
[0, 1]	Hydrogen-bond donor + hydrogen-bond acceptor
[0, 2]	Hydrogen-bond donor + positive charge
[0, 3]	Hydrogen-bond donor + negative charge
[1, 2]	Hydrogen-bond acceptor + positive charge
[1, 3]	Hydrogen-bond acceptor + negative charge
[2, 3]	Positive charge + negative charge
[0, 1, 2]	Hydrogen-bond donor + hydrogen-bond acceptor + positive charge
Edge attributes	
Attribute	**Description**
-	Single bond
=	Double bond
≡	Triple bond

Table 3. Substitution, insertion and deletion costs for nodes, as proposed by Harper *et al.* [38].

Substitution Costs for Nodes															
	[0]	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**	**[7]**	**[0, 1]**	**[0, 2]**	**[0, 3]**	**[1, 2]**	**[1, 3]**	**[2, 3]**	**[0, 1, 2]**
[0]	0	2	2	2	2	2	2	3	1	1	1	2	2	2	1
[1]	2	0	2	2	2	2	2	3	1	2	2	1	1	2	1
[2]	2	2	0	2	2	2	2	3	2	1	2	1	2	1	1
[3]	2	2	2	0	2	2	2	3	2	2	1	2	1	1	2
[4]	2	2	2	2	0	2	2	3	2	2	2	2	2	2	2
[5]	2	2	2	2	2	0	2	3	2	2	2	2	2	2	2
[6]	2	2	2	2	2	2	0	3	2	2	2	2	2	2	2
[7]	3	3	3	3	3	3	3	0	3	3	3	3	3	3	3
[0, 1]	1	1	2	2	2	2	2	3	0	2	2	2	2	2	2
[0, 2]	1	2	1	2	2	2	2	3	2	0	2	2	2	2	2
[0, 3]	1	2	2	1	2	2	2	3	2	2	0	2	2	2	2
[1, 2]	2	1	1	2	2	2	2	3	2	2	2	0	2	2	2
[1, 3]	2	1	2	1	2	2	2	3	2	2	2	2	0	2	2
[2, 3]	2	2	1	1	2	2	2	3	2	2	2	2	2	0	2
[0, 1, 2]	1	1	1	2	2	2	2	3	2	2	2	2	2	2	0
Insertion/Deletion Costs for Nodes															
	[0]	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**	**[7]**	**[0, 1]**	**[0, 2]**	**[0, 3]**	**[1, 2]**	**[1, 3]**	**[2, 3]**	**[0, 1, 2]**
insert	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2
delete	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2

Table 4. Substitution, insertion and deletion costs for edges, as proposed by Harper *et al.* [38].

Substitution Costs for Edges			
	-	=	≡
-	0	3	3
=	3	0	3
≡	3	3	0
Insertion/Deletion Costs for Edges			
	-	=	≡
Insert	0	1	1
Delete	0	1	1

The final edit cost for a given edit path is obtained by adding up all individual transformation costs. Fig. (**3**) shows an example of edit path. In this case, the transformation sequence is the sum for the cost of deleting an edge, plus the cost of deleting node 3, plus the cost of inserting node 2, plus the cost of inserting an edge, plus the cost of substitution from node 1 to node 4. The process of adding up all transformation costs and getting the edit cost is repeated for any possible edit path transforming one graph into the other. At the end, the GED resulting value for any pair of graphs *A* and *B* is defined as the minimum cost under all those possible edit path sequences.

Usually, the GED is normalized according to the number of nodes in both graphs being compared. This is done in order to make the measure independent of the size of the graphs.

Several GED computational methods have been proposed during the last three decades, and they can be classified into two groups: those returning the exact value for the GED in exponential time proportional to the number of nodes [56]; and those returning an approximation of the GED in polynomial time [57, 58]. These two groups of GED computational methods have been widely studied [59, 60].

In this work, we computed an approximation of the GED in polynomial time using an in-house implementation of the bipartite graph matching method proposed by Serratosa [57]. This was programmed in C++ and Python languages.

2.4. Objective Function

Fig. (**4**) shows the objective function we have used along with a genetic algorithm [61, 62] to track the evolution of the

learning process. The objective function is divided into three main steps. First, we compare each molecule with all the others and create a matrix of distances. Second, for each row in the matrix (which represents the distances computed for one molecule, the "query molecule", with all the others), we find the lowest distance D, which is considered as the "closest molecule" to the query one. Third, we use the "closest molecule" distance D and a log loss function [63] adding up the quantity $exp(-D)$ as follows: if the "query molecule" and the "closest molecule" are from the same activity class, then the objective function is decreased. On the contrary, if the "query molecule" and the "closest molecule" are from different activity classes, then the objective function is increased. The log loss uses the error magnitude in the prediction (how much it varies from the ground truth) to give a more continuous view over the model's behavior, slowly increasing or decreasing the objective function output values. Note that for building the matrix of distances, if the molecules belong to the same activity class, then the lower the distance the better. On the other hand, if the molecules belong to different activity classes, then the higher the distance the better.

The learning algorithm tries to minimize the objective function, therefore it ends up having as many correct classifications as possible, since correct classifications reduce the resulting value and wrong classifications increase it, as explained before. The main goal of this process is to measure the performance of the objective function and tune the next values to be used as edit costs.

3. EXPERIMENTAL

The aim of the practical experiments is twofold. First, we want to compare the recognition ratio deduced using the learned edit costs to the recognition ratio deduced using the edit costs proposed by Harper *et al.* [38]. Second, we want to analyze if the learned costs are congruent with the chemical knowledge given by Harper *et al.* [38]. To that aim, we have done four experiments and, in each experiment, we have learned one edit cost. The edit costs have been selected considering the more frequent node and edge attributes.

- **Experiment 1:** insertion and deletion costs corresponding to the carbon link node, which is assigned to attribute "[6]" in Table **2**.
- **Experiment 2:** substitution cost between the carbon link node (attribute "[6]" in Table **2**) and the aromatic ring system node (attribute "[5]" in Table **2**).
- **Experiment 3:** insertion and deletion costs corresponding to the single bond edge, which is assigned to attribute "-" in Table **2**.
- **Experiment 4:** substitution cost between the single bond edge (attribute "-" in Table **2**) and the double bond edge (attribute "=" in Table **2**).

4. RESULTS

Fig. (**5**) shows an example of the learning behavior of a single target. In this figure, the blue continuous line represents the objective function, which decreases after every learning iteration until convergence. The red-segmented line represents the number of misclassifications over the training set, and the green points represent the number of misclassifications over the test set. Both training and test misclassification values should decrease, but it is not always the case because it depends on several factors including the size of the training set, the number of variables being learned, the tolerance for convergence and the overfitting, among others.

Fig. (**6**) shows the number of errors in the classification process over the test set for all 127 targets, using two different edit cost configurations: the edit costs proposed by Harper *et al.* [38], and the edit costs we have learned. It is important to note that, since the figure depicts the number of misclassifications, the lower the values the better. These results show how the learned edit costs present a slightly improved behavior compared to the edit costs proposed by Harper *et al.* [38]. The improvement is noted in the maximum and the third quartile being part of the box-and-whisker plots. All other quartile values are the same for both methods.

```
input: learning_set,
       edit_costs
output: F

begin
    # step 1: compute the matrix of distances
    ∀ Gi in learning_set
        ∀ Gj in learning_set
            Dist (i , j) = GED(Gi , Gj , edit_costs)
        end∀
    end∀
    # initialize the resulting value
    F = 0
    # for each row in the matrix of distances
    ∀ Gi in learning_set
        # step 2: find the closest molecule and its index
        D = min∀j {Dist (i , j)}
        K = argmin∀j {Dist (i , j)}
        # step 3: add up or subtract depending on classification
        if class (GK) = class (Gi)
            F = F - e-D
        else
            F = F + e-D
        end if
    end∀
end
```

Fig. (4). Objective function.

We present in Fig. (**7**), a deeper analysis for each dataset separately in order to better understand the behavior of the learned costs. This figure shows the number of errors in the classification for two datasets in each experiment. (We do not include all the results per experiment for space reasons; nevertheless, other results can be found as supplementary material). Note that each row represents an experiment and each subfigure in the row represents a dataset. As in Fig. (**6**), here we show the number of misclassifications. Again, box-and-whisker plots are located on the right of each subplot to illustrate the distribution of values.

For the first experiment, we used the GLL&GDD and DUD-E datasets. The results using the learned edit costs are slightly better. The improvement obtained by using the learned costs can be observed in the third quartile and the maximum value for the GLL&GDD dataset, and in the median and maximum value for the DUD-E dataset. For the GLL&GDD, as the third quartile is reduced, note how the GED using the learned costs provides more stable results (Stability is represented as the box and whiskers length; shorter lengths indicate that several results are closer to each other, so the method seems more reliable). For the other datasets in this experiment, results were similar, obtaining lower or equal median values using the learned edit costs as compared to the edit costs from Harper *et al.* [38].

For the second experiment, we used the ULS-UDS and GLL&GDD datasets. In this case, the results using the learned edit costs are significantly better, too. The improvement obtained by using the learned costs can be observed in every aspect of the box plot for the ULS-UDS dataset and in almost every aspect of the box plot for the GLL&GDD

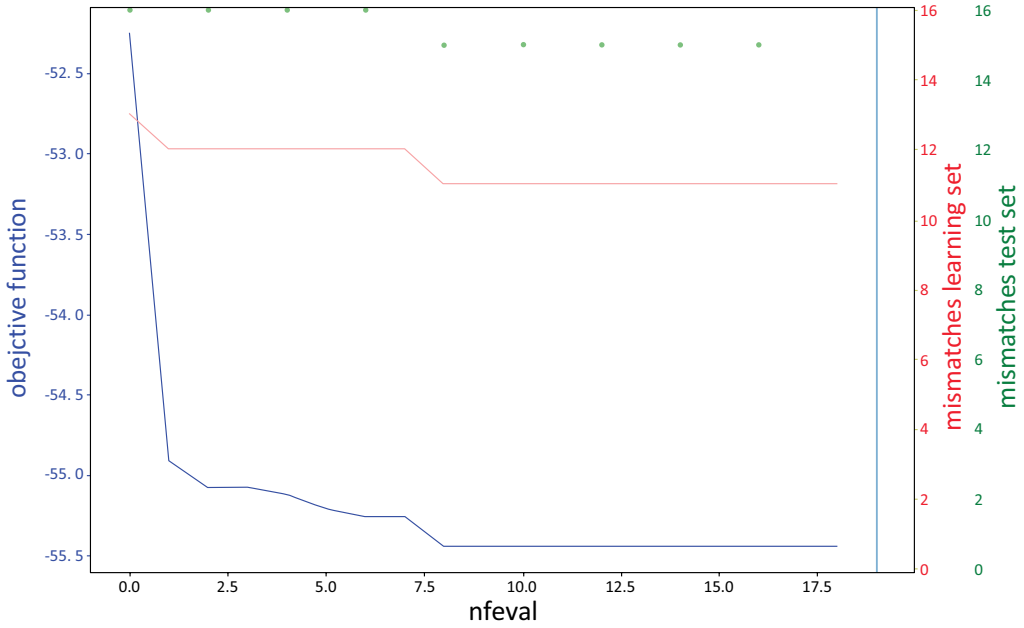

Fig. (5). Training evolution for target FXA in dataset DUD-E.

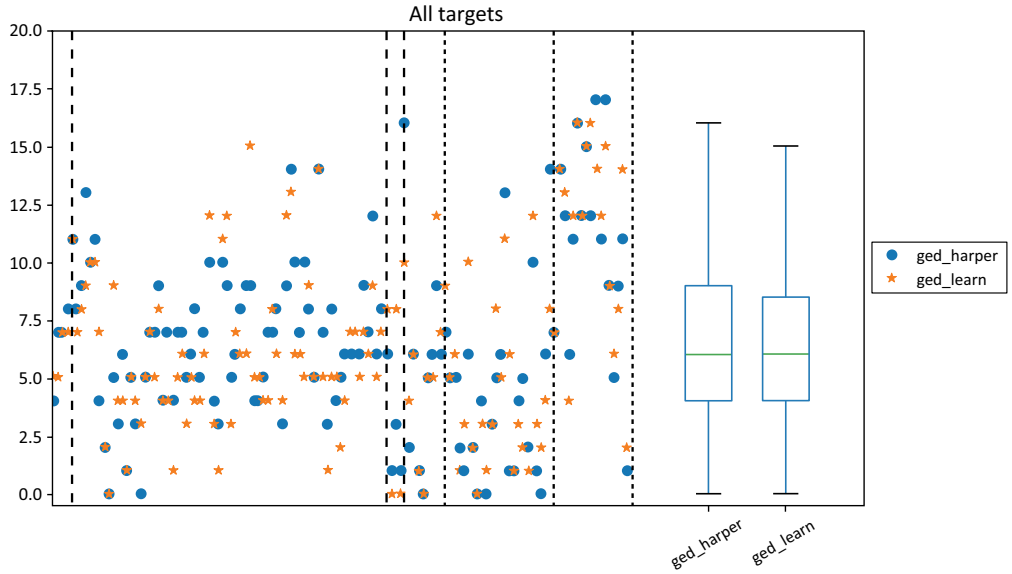

Fig. (6). Number of misclassifications using the test set over the 127 targets available in the six datasets combined. The scattered values on the left of the plot represent the number of classification errors (the lower the values, the better) using different colors and shapes depending on the edit costs used. Vertical segmented lines mark the limits between different datasets (from left to right: ULS-UDS, GLL&GDD, CAPST, DUD-E, NRLiSt_BDB, and MUV). The box-and-whisker plots on the right show the distribution of the resulting values. The boxes show the first and third quartiles, the line in the middle of the box is the median value (second quartile), and the whiskers extend from the boxes to show the range of the data (outliers are not included).

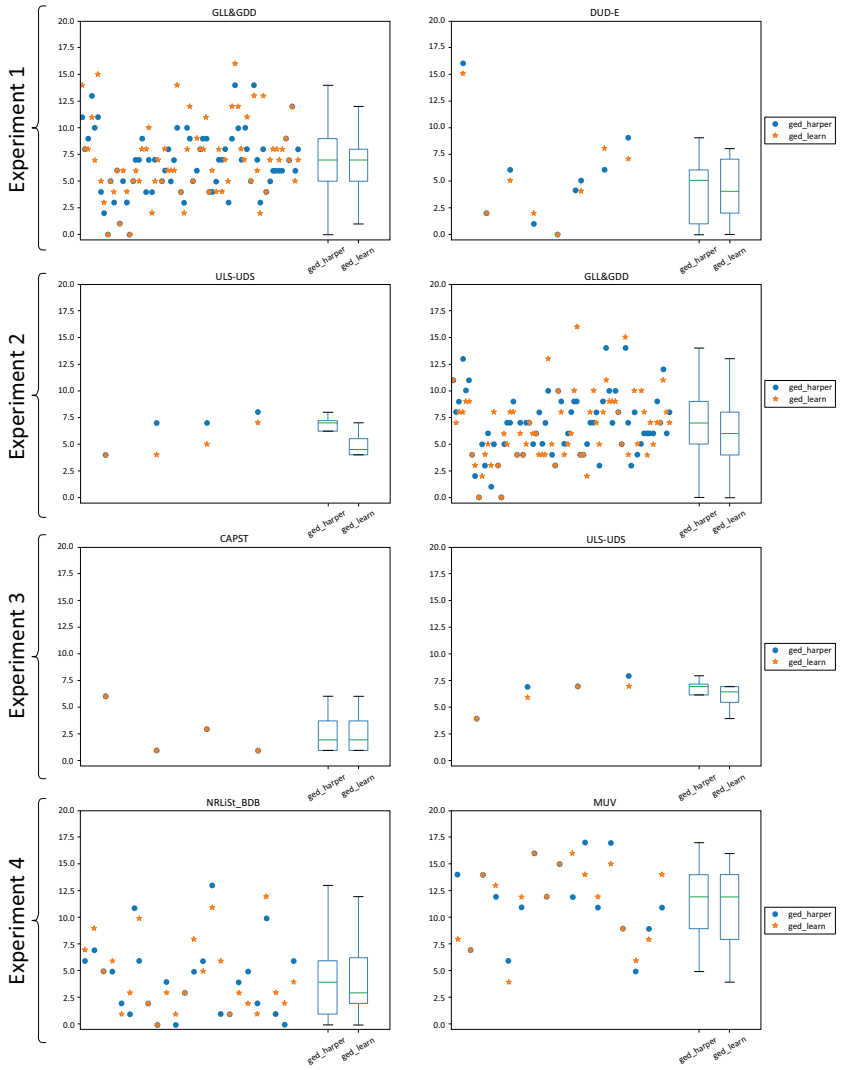

Fig. (7). The number of misclassifications for all available targets in the LBVS benchmarking platform separated for each dataset and each experiment. The scattered values on the left of each subplot represent the number of classification errors (the lower the values, the better) using different colors and shapes depending on the edit costs used. Box-and-whisker plots on the right of each subplot show the distribution of the resulting values for each experiment.

dataset, except the minimum value, which is zero in both cases. For the other datasets in this experiment, median values using the learned and Harper's edit costs were the same except for CAPST, which obtained a better median value using Harper's costs.

For the third experiment, we used CAPST and ULS-UDS datasets. In this case, only the results for ULS-UDS using the learned edit costs are better. This improvement is noticeable for all the values in the box plot, including the three quartiles and the minimum and maximum values. On the other hand, for the CAPST dataset, the values are the same for each target using the learned edit costs and the edit costs proposed by Harper. For the other datasets in this experiment, median values using learned and Harper's edit costs were the same in everycase.

Finally, for the fourth experiment we used the NRLiSt_BDB and MUV datasets. In this experiment, the results using learned edit costs are slightly better. The improvement obtained by using learned costs can be observed in the median and maximum values for the NRLiSt_BDB dataset, while for the MUV dataset the improvement can be observed in the first quartile, minimum and maximum values. Nevertheless, the third quartile is better for the NRLiSt_BDB dataset using the costs proposed by Harper. For the other datasets in this experiment, results were similar, obtaining lower or equal median values using learned edit costs as compared to the edit costs from Harper, except for CAPST, which obtained a better median value using Harper's costs.

5. DISCUSSION

GED along with ErGs using learned edit costs obtained better recognition ratio results in most experiments. In this experiments, we only learned one edit cost per case, and we chose this simpler and clearer approach to show the validity of our method. Clearly, this methodology can be applied to learn several edit costs at a time, either sequentially or in par-

Table 5. Harper's costs and learned values obtained per experiment.

	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Harper *et al.*	1	2	0	3
CAPST	0.000	0.013	0.004	0.017
DUD-E	0.005	0.145	0.001	0.186
GLL&GDD	0.014	0.333	0.003	0.206
MUV	0.490	0.867	0.327	1.005
NRLiSt_BDB	0.012	0.104	0.003	0.024
ULS-UDS	0.115	0.500	0.011	0.607

allel. Learning a bigger number of edit costs might increase the recognition ratio values with respect to those presented in this study.

Table **5** shows the edit costs proposed by Harper *et al.* [38] and the learned edit costs obtained for each one of the experiments. We can see how the learned edit costs of the first experiment tend to be lower than Harper's edit costs in most datasets. This means that, in general, inserting or deleting a link node should imply a lower cost than the expected by Harper *et al.* [38]. On the other hand, for the third experiment, learned edit costs tend to be slightly higher than Harper's edit costs, meaning that inserting or deleting a single bond edge should imply a higher cost than the expected by Harper *et al.* [38]. Furthermore, learned edit costs for the second and the fourth experiments tend to be greater compared to the first and the third experiments, which means that the substitution of a link node for an aromatic ring node or the substitution of a single bond edge for a double bond edge should carry a higher cost than inserting or deleting a link node or a single bond edge. This information is useful in order to have some clues about the structure-activity relationship within molecules.

In a previous work [39], we presented a molecular similarity measure that uses graph edit distance to effectively compare the representation of molecules by extended reduced graphs. In that work, edit costs for the different node and edge operations were assigned using expert knowledge. In this study, we have used a learning algorithm to learn the edit costs automatically. Significant improvements in performance have been obtained when using learned costs in most of the experiments for the 127 targets present in six datasets. All datasets used are publicly available as part of the benchmarking platform proposed by Skoda and Hoksza [53].

CONCLUSION

Results show that learned edit costs performed as good or better in most of the targets present in the six datasets, compared to the edit costs proposed by Harper *et al.* [38]. In addition, learned costs may also give some ideas related to the structure-activity relation present within activity classes.

Further research will focus on comparing the results in this study with other results obtained by learning a greater number of edit costs (for instance, including all insertion and deletion costs or all substitution costs), and a variety of toxicological endpoints.

For purposes of simplicity, the GED implementation used in this study does not envisage the use of stereochemical information for molecules. This issue could be addressed in future studies. It should be possible to include this information since the 3D location of each atom is available for all the datasets in the LBVS benchmarking platform, so a reference for the position of the neighbors with respect to each atom should be possible to be established.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No animals/humans were used for studies that are base of this research.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

Six publicly available datasets were used in this study. They are: ULS and UDS [47], GDD and GLL [48], DUD Release 2 (DUD-E) [49], NRLiSt BDB [50], MUV [51], and a dataset from Comparative Analysis of Pharmacophore Screening Tools (CAPST) [52].

FUNDING

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713679 and from the Universitat Rovira i Virgili (URV). This study has also been partially supported by the Spanish projects TIN2016-77836-C2-1-R and ColRobTransp MINECO DPI2016-78957-R AEI/FEDER EU.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Carlos Garcia-Hernandez collected data, performed research, analyzed results and wrote the paper. Alberto Fernández designed the research, analyzed results and wrote the paper. Francesc Serratosa designed research, analyzed results and wrote the paper.

SUPPLEMENTARY MATERIAL

Supplementary material is available on the publisher's web site along with the published article.

REFERENCES

[1] Kubinyi, H.; Mannhold, R.; Timmerman, H. *Virtual screening for bioactive molecules*; John Wiley & Sons: Hoboken, **2008**, Vol. *10*.

[2] Schneider, G.; Clément-Chomienne, O.; Hilfiger, L.; Schneider, P.; Kirsch, S.; Böhm, H.J.; Neidhart, W. Virtual screening for bioactive molecules by evolutionary de novo design. *Angew. Chem. Int. Ed. Engl.,* **2000**, *39*(22), 4130-4133. http://dx.doi.org/10.1002/1521-3773(20001117)39:22<4130::AID-ANIE4130>3.0.CO;2-E PMID: 11093229

[3] Bajorath, J. Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput. Sci.,* **2001**, *41*(2), 233-245. http://dx.doi.org/10.1021/ci0001482 PMID: 11277704

[4] Heikamp, K.; Bajorath, J. The future of virtual compound screening. *Chem. Biol. Drug Des.,* **2013**, *81*(1), 33-40. http://dx.doi.org/10.1111/cbdd.12054 PMID: 23253129

[5] Cereto-Massagué, A.; Ojeda, M.J.; Valls, C.; Mulero, M.; Garcia-Vallvé, S.; Pujadas, G. Molecular fingerprint similarity search in virtual screening. *Methods,* **2015**, *71*, 58-63. http://dx.doi.org/10.1016/j.ymeth.2014.08.005 PMID: 25132639

[6] Kroemer, R.T. Structure-based drug design: docking and scoring. *Curr. Protein Pept. Sci.,* **2007**, *8*(4), 312-328. http://dx.doi.org/10.2174/138920307781369382 PMID: 17696866

[7] Cavasotto, C.N.; Orry, A.J.; Andrew, J. Ligand docking and structure-based virtual screening in drug discovery. *Curr. Top. Med. Chem.,* **2007**, *7*(10), 1006-1014. http://dx.doi.org/10.2174/156802607780906753 PMID: 17508934

[8] Sun, H. Pharmacophore-based virtual screening. *Curr. Med. Chem.,* **2008**, *15*(10), 1018-1024. http://dx.doi.org/10.2174/092986708784049630 PMID: 18393859

[9] Kirchmair, J.; Distinto, S.; Markt, P.; Schuster, D.; Spitzer, G.M.; Liedl, K.R.; Wolber, G. How to optimize shape-based virtual screening: choosing the right query and including chemical information. *J. Chem. Inf. Model.,* **2009**, *49*(3), 678-692. http://dx.doi.org/10.1021/ci8004226 PMID: 19434901

[10] Melville, J.L.; Burke, E.K.; Hirst, J.D. Machine learning in virtual screening. *Comb. Chem. High Throughput Screen.,* **2009**, *12*(4), 332-343. http://dx.doi.org/10.2174/138620709788167980 PMID: 19442063

[11] Johnson, M.A.; Maggiora, G.M. *Concepts and applications of molecular similarity*; Wiley & Sons: Hoboken, **1990**.

[12] Bender, A.; Glen, R.C. Molecular similarity: a key technique in molecular informatics. *Org. Biomol. Chem.,* **2004**, *2*(22), 3204-3218. http://dx.doi.org/10.1039/b409813g PMID: 15534697

[13] Nikolova, N.; Jaworska, J. Approaches to measure chemical similarity - a review. *Mol. Inform.,* **2003**, *22*, 1006-1026.

[14] Willett, P. Evaluation of molecular similarity and molecular diversity methods using biological activity data. In: *Cheminformatics;* Springer: Berlin, **2004**, p.p. 51-63. http://dx.doi.org/10.1385/1-59259-802-1:051

[15] Lajiness, M. *Molecular similarity-based methods for selecting compounds for screening. Computational chemical graph theory*; Nova Science Publishers, Inc., **1990**, pp. 299-316.

[16] Willett, J. *Similarity and clustering in chemical information systems*; John Wiley & Sons, Inc.: Hoboken, **1987**.

[17] Sheridan, R.P.; Kearsley, S.K. Why do we need so many chemical similarity search methods? *Drug Discov. Today,* **2002**, *7*(17), 903-911. http://dx.doi.org/10.1016/S1359-6446(02)02411-X PMID: 12546933

[18] Willett, P.; Barnard, J.M.; Downs, G.M. Chemical similarity searching. *J. Chem. Inf. Comput. Sci.,* **1998**, *38*, 983-996. http://dx.doi.org/10.1021/ci9800211

[19] Xue, L.; Bajorath, J. Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Comb. Chem. High Throughput Screen.,* **2000**, *3*(5), 363-372. http://dx.doi.org/10.2174/1386207003331454 PMID: 11032954

[20] Menard, P.R.; Mason, J.S.; Morize, I.; Bauerschmidt, S. Chemistry space metrics in diversity analysis, library design, and compound selection. *J. Chem. Inf. Comput. Sci.,* **1998**, *38*, 1204-1213. http://dx.doi.org/10.1021/ci9801062

[21] Pearlman, R.S.; Smith, K.M. Metric validation and the receptor-relevant subspace concept. *J. Chem. Inf. Comput. Sci.,* **1999**, *39*, 28-35. http://dx.doi.org/10.1021/ci980137x

[22] Schnur, D. Design and diversity analysis of large combinatorial libraries using cell-based methods. *J. Chem. Inf. Comput. Sci.,* **1999**, *39*, 36-45. http://dx.doi.org/10.1021/ci980138p

[23] Livingstone, D.J. The characterization of chemical structures using molecular properties. A survey. *J. Chem. Inf. Comput. Sci.,* **2000**, *40*(2), 195-209. http://dx.doi.org/10.1021/ci990162i PMID: 10761119

[24] Barnard, J.M. Substructure searching methods: Old and new. *J. Chem. Inf. Comput. Sci.,* **1993**, *33*, 532-538. http://dx.doi.org/10.1021/ci00014a001

[25] James, C.; Weininger, D. *Daylight, 4.41 Theory Manual*; Daylight Chemical Information Systems Inc.: Irvine, CA, USA, **1995**.

[26] MACCS. *K. MDL Information Systems. Inc*; San Leandro: CA, **1984**.

[27] McGregor, M.J.; Pallai, P.V. Clustering of large databases of compounds: using the MDL "keys" as structural descriptors. *J. Chem. Inf. Comput. Sci.,* **1997**, *37*, 443-448. http://dx.doi.org/10.1021/ci960151e

[28] Güner, O.F. Pharmacophore perception, development and use in drug design. *Molecules,* **2000**, *5*(7), 987-989.

[29] Beno, B.R.; Mason, J.S. The design of combinatorial libraries using properties and 3D pharmacophore fingerprints. *Drug Discov. Today,* **2001**, *6*(5), 251-258. http://dx.doi.org/10.1016/S1359-6446(00)01665-2 PMID: 11182598

[30] Rarey, M.; Dixon, J.S. Feature trees: a new molecular similarity measure based on tree matching. *J. Comput. Aided Mol. Des.,* **1998**, *12*(5), 471-490. http://dx.doi.org/10.1023/A:1008068904628 PMID: 9834908

[31] Takahashi, Y.; Sukekawa, M.; Sasaki, S. Automatic identification of molecular similarity using reduced-graph representation of chemical structure. *J. Chem. Inf. Model.,* **1992**, *32*, 639-643. http://dx.doi.org/10.1021/ci00010a009

[32] Barker, E.J.; Buttar, D.; Cosgrove, D.A.; Gardiner, E.J.; Kitts, P.; Willett, P.; Gillet, V.J. Scaffold hopping using clique detection applied to reduced graphs. *J. Chem. Inf. Model.,* **2006**, *46*(2), 503-511. http://dx.doi.org/10.1021/ci050347r PMID: 16562978

[33] Fisanick, W.; Lipkus, A.H.; Rusinko, A. Similarity searching on CAS registry substances. 2. 2D structural similarity. *J. Chem. Inf. Comput. Sci.,* **1994**, *34*, 130-140. http://dx.doi.org/10.1021/ci00017a016

[34] Gillet, V.J.; Downs, G.M.; Holliday, J.D.; Lynch, M.F.; Dethlefsen, W. Computer storage and retrieval of generic chemical structures in patents. 13. Reduced graph generation. *J. Chem. Inf. Comput. Sci.,* **1991**, *31*, 260-270. http://dx.doi.org/10.1021/ci00002a011

[35] Gillet, V.J.; Willett, P.; Bradshaw, J. Similarity searching using reduced graphs. *J. Chem. Inf. Comput. Sci.,* **2003**, *43*(2), 338-345. http://dx.doi.org/10.1021/ci025592e PMID: 12653495

[36] Stiefl, N.; Watson, I.A.; Baumann, K.; Zaliani, A. ErG: 2D pharmacophore descriptions for scaffold hopping. *J. Chem. Inf. Model.,* **2006**, *46*(1), 208-220.

http://dx.doi.org/10.1021/ci050457y PMID: 16426057

[37] Barker, E.J.; Gardiner, E.J.; Gillet, V.J.; Kitts, P.; Morris, J. Further development of reduced graphs for identifying bioactive compounds. *J. Chem. Inf. Comput. Sci.,* **2003**, *43*(2), 346-356. http://dx.doi.org/10.1021/ci0255937 PMID: 12653496

[38] Harper, G.; Bravi, G.S.; Pickett, S.D.; Hussain, J.; Green, D.V.S. The reduced graph descriptor in virtual screening and data-driven clustering of high-throughput screening data. *J. Chem. Inf. Comput. Sci.,* **2004**, *44*(6), 2145-2156. http://dx.doi.org/10.1021/ci049860f PMID: 15554685

[39] Garcia-Hernandez, C.; Fernández, A.; Serratosa, F. Ligand-based virtual screening using graph edit distance as molecular similarity measure. *J. Chem. Inf. Model.,* **2019**, *59*(4), 1410-1421. http://dx.doi.org/10.1021/acs.jcim.8b00820 PMID: 30920214

[40] Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.,* **1957**, *5*, 32-38. http://dx.doi.org/10.1137/0105003

[41] Sanfeliu, A.; Fu, K.S. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Syst. Man Cybern.,* **1983**, 353-362. http://dx.doi.org/10.1109/TSMC.1983.6313167

[42] Gao, X.; Xiao, B.; Tao, D.; Li, X. A survey of graph edit distance. *Pattern Anal. Appl.,* **2010**, *13*, 113-129. http://dx.doi.org/10.1007/s10044-008-0141-y

[43] Solé, A.; Serratosa, F.; Sanfeliu, A. On the graph edit distance cost: properties and applications. *Int. J. Pattern Recognit. Artif. Intell.,* **2012**, *26,* 1260004. http://dx.doi.org/10.1142/S021800141260004X

[44] Verma, J.; Khedkar, V.M.; Coutinho, E.C. 3D-QSAR in drug design--a review. *Curr. Top. Med. Chem.,* **2010**, *10*(1), 95-115. http://dx.doi.org/10.2174/156802610790232260 PMID: 19929826

[45] Lewis, R.A. A general method for exploiting QSAR models in lead optimization. *J. Med. Chem.,* **2005**, *48*(5), 1638-1648. http://dx.doi.org/10.1021/jm049228d PMID: 15743205

[46] Birchall, K.; Gillet, V.J.; Harper, G.; Pickett, S.D. Training similarity measures for specific activities: application to reduced graphs. *J. Chem. Inf. Model.,* **2006**, *46*(2), 577-586. http://dx.doi.org/10.1021/ci050465e PMID: 16562986

[47] Xia, J.; Tilahun, E.L.; Reid, T.E.; Zhang, L.; Wang, X.S. Benchmarking methods and data sets for ligand enrichment assessment in virtual screening. *Methods,* **2015**, *71*, 146-157. http://dx.doi.org/10.1016/j.ymeth.2014.11.015 PMID: 25481478

[48] Gatica, E.A.; Cavasotto, C.N. Ligand and decoy sets for docking to G protein-coupled receptors. *J. Chem. Inf. Model.,* **2012**, *52*(1), 1-6. http://dx.doi.org/10.1021/ci200412p PMID: 22168315

[49] Mysinger, M.M.; Carchia, M.; Irwin, J.J.; Shoichet, B.K. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.,* **2012**, *55*(14), 6582-6594. http://dx.doi.org/10.1021/jm300687e PMID: 22716043

[50] Lagarde, N.; Ben Nasr, N.; Jérémie, A.; Guillemain, H.; Laville, V.; Labib, T.; Zagury, J.F.; Montes, M. NRLiSt BDB, the manually curated nuclear receptors ligands and structures benchmarking database. *J. Med. Chem.,* **2014**, *57*(7), 3117-3125. http://dx.doi.org/10.1021/jm500132p PMID: 24666037

[51] Rohrer, S.G.; Baumann, K. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *J. Chem. Inf. Model.,* **2009**, *49*(2), 169-184. http://dx.doi.org/10.1021/ci8002649 PMID: 19434821

[52] Sanders, M.P.; Barbosa, A.J.; Zarzycka, B.; Nicolaes, G.A.; Klomp, J.P.; de Vlieg, J.; Del Rio, A. Comparative analysis of pharmacophore screening tools. *J. Chem. Inf. Model.,* **2012**, *52*(6), 1607-1620. http://dx.doi.org/10.1021/ci2005274 PMID: 22646988

[53] Skoda, P.; Hoksza, D. In: *Benchmarking platform for ligand-based virtual screening.* Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Shenzhen, China, **2016**, pp. 1220-1227. http://dx.doi.org/10.1109/BIBM.2016.7822693

[54] Riniker, S.; Landrum, G.A. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *J. Cheminform.,* **2013**, *5*(1), 26. http://dx.doi.org/10.1186/1758-2946-5-26 PMID: 23721588

[55] Kearsley, S.K.; Sallamack, S.; Fluder, E.M.; Andose, J.D.; Mosley, R.T.; Sheridan, R.P. Chemical similarity using physiochemical property descriptors. *J. Chem. Inf. Comput. Sci.,* **1996**, *36*, 118-127. http://dx.doi.org/10.1021/ci950274j

[56] Blumenthal, D.B.; Gamper, J. On the exact computation of the graph edit distance. *Pattern Recognit. Lett.,* **2018**, *134*, 1-12. http://dx.doi.org/10.1016/j.patrec.2018.05.002

[57] Serratosa, F. Fast computation of bipartite graph matching. *Pattern Recognit. Lett.,* **2014**, *45*, 244-250. http://dx.doi.org/10.1016/j.patrec.2014.04.015

[58] Santacruz, P.; Serratosa, F. Error-tolerant graph matching in linear computational cost using an initial small partial matching. *Pattern Recognit. Lett.,* **2018**, *134,* 10-19. http://dx.doi.org/10.1016/j.patrec.2018.04.003

[59] Conte, D.; Foggia, P.; Sansone, C.; Vento, M. Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognit. Artif. Intell.,* **2004**, *18*, 265-298. http://dx.doi.org/10.1142/S0218001404003228

[60] Vento, M. A long trip in the charming world of graphs for pattern recognition. *Pattern Recognit.,* **2015**, *48*, 291-301. http://dx.doi.org/10.1016/j.patcog.2014.01.002

[61] Goldberg, D.E.; Holland, J.H. Genetic algorithms and machine learning. *Mach. Learn.,* **1988**, *3*, 95-99. http://dx.doi.org/10.1023/A:1022602019183

[62] Storn, R.; Price, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.,* **1997**, *11*, 341-359. http://dx.doi.org/10.1023/A:1008202821328

[63] Rosasco, L.; De Vito, E.; Caponnetto, A.; Piana, M.; Verri, A. Are loss functions all the same? *Neural Comput.,* **2004**, *16*(5), 1063-1076. http://dx.doi.org/10.1162/089976604773135104 PMID: 15070510