MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **55** (2006) 205-232

ISSN 0340 - 6253

New Molecular Fragmental Descriptors and Their Application to the Prediction of Fish Toxicity

Sokratis Alikhanidi* and Yoshimasa Takahashi

Department of Knowledge-based Information Engineering,

Toyohashi University of Technology, Tempaku-cho, Toyohashi 441-8580, Japan

*e-mail: socrates@mail.ru

(Received August 5, 2004)

Abstract

The new general concept of the molecular generalized (fuzzy) fragments is introduced and the whole algorithm is presented in details. The fragment generation is highly flexible and depends on several collections of rules. All molecular fragments can be distinguished as trivial or non-trivial. The latter have been named the Fuzzy Characteristic Groups (FCG), since they only consist of the non-trivial, possibly fuzzy, molecular bonds. The developed new fragmental descriptors were applied to QSAR modeling of fish toxicity for the structurally diverse data set of 478 molecules. The found PLS (partial least squares) model, which was successfully validated by 94 molecules from the test set, showed the good stability of the model on the unknown data.

Introduction

The preliminary evaluation of certain molecular properties without the direct synthesis of a compound plays very important role in the modern organic chemistry. Such evaluation dramatically decreases the cost of new compounds development by suppressing theoretically the unsuccessful candidates, which otherwise must be prepared and tested experimentally. The Quantitative Structure-Activity Relationship (QSAR) is a very powerful way for estimation and improvement of different molecular properties [1]. According to the method, a property is described in a mathematical manner (equation, logic, more sophisticated models) using calculable descriptors – invariants of molecular structure. The mathematical basis of the approach is commonly acquired from the Data Mining science [2]. The molecular descriptors represent another essential part of QSAR analysis. They may be classified into different categories [3]: empirical (induction, resonance, and steric constants; polarity, polarizability, acidity, basicity, and mixed scales), constitutional (counts of atoms and bonds, molecular weight), topological (graph-based indices), geometrical (distance, surface-area, volume, and their related parameters), quantum chemical descriptors (atomic partial charges, bond orders), etc. The calculated or experimentally observed physicochemical properties (octanol-water partition coefficient, refractivity, polarizability) and the frequencies (or just presence) of particular functional groups or fragments are also frequently employed.

The important desired feature of molecular descriptors is their clear meaning – “transparency”, which provides easier interpretation for the whole QSAR model [4]. Many of the topological and geometrical descriptors lack this characteristic. Besides, many global molecular descriptors may often be considered as redundant, and cannot be applied successfully to a particular QSAR problem [5]. In many cases, especially when a data set consists of molecules of different structural classes, introducing the descriptors that depict the local structural features of this collection may improve the model.

Probably the most classical structural invariant is the presence of specific functional groups looking friendly for a chemist's eye. This fundamental and natural type of molecular descriptors has been successfully used for development of the QSAR models for lipophilicity [6], solubility [7], and biodegradability [8]. Fragmental descriptors have also been used for clustering of the structurally diverse chemicals followed by the development of the local QSAR models for structurally similar molecules [9,10]. The recent review on the fragmental QSAR models has highlighted other application instances [11].

Method

Very often chemists distinguish different forms of the same atom – depending on its electronic state and neighbors. As example, carbon atoms in carboxyl group and methyl group differ greatly; hydrogen atoms of phenyl, hydroxyl, or methyl groups are also significantly different. Such separation is sometimes carried out using an alternative definition of atom types (AT) [12,13]. On the other hand, atoms with different predefined AT or even different elements may have some similar features. Halogens are the typical case (chlorine and bromine are very often discussed in the literature together). Saturated carbon, ester oxygen, amine nitrogen, and sulfide sulfur may be considered as the conventional skeleton's atoms. Nucleophiles, electrophiles, heavy atoms, and heteroatoms are other well-known aggregations of the significantly different atoms. This list can be easily enlarged specifically as the premise for modeling a molecular property of the interest.

In this paper, we present the flexible method for generation of molecular fragments with the user-defined rules for assigning the atom types and the rules for atom and bond generalization. To suppress a lot of generated redundant fragments, a list of trivial bonds is used; any such bond in a fragment marks it to be the trivial fragment. This fragment will not be used as a pattern on the next generalization step, and can be removed from the output

fragment list. On the other hand, the non-trivial fragment bears the valuable information on the molecular functionality, and may play an important role in QSAR analysis. Thus, we suggest defining the non-trivial fragments as the Characteristic Groups of a molecule.

Our approach consists of several algorithmic parts as follows.

1. Assigning all atoms to their atom types (optional).
2. Generation of all the molecular fragments of the specified size suppressing their duplication followed by marking the trivial fragments.
3. Generalization step: do the fuzzy matching of fragments according to the user-specified rules. Match one fragment ("pattern", only non-trivial) over another ("target", any fragment) of the same topology to get "templates" of the same topology (neither atom nor bond miss is allowed). Templates are combined by logical addition to produce all possible combinations without repetition.
4. Counting of fragments in each molecule (optional).

Atom types (AT). AT depends on the periodic system element name of the atom, presence of the atom in rings of specific size, element names of the first-order neighbors, and types of bonds to the first-order neighbors. The rules are written in a text file with the hierarchical decision structure:

Element:

Tests?

Assignment statement.

There are two types of tests:

in n ring? n is in the range 3-6.

or *connected to <expression>?*

The <*expression*> statement has rich syntax based on the Perl regular expressions [14], and provides flexible checking the nearest neighbors of the atom (see Fig. 1) for presence/absence of the specific element types in different combinations together with the bond types. The scanning is from top to bottom and from left to right, so the order of clauses is important. For multiple tests or expressions, each one must be satisfied to finalize the corresponding type.

The choice of AT rules and its application are options that offer to discriminate naturally different forms of atoms. While the user can omit this step to prefer the ordinary element names from the periodic system, the usage of AT provides a convenient way to keep good informational filling of even the short fragments.

Table 1. The set of atom types used in this analysis.

ACT	Explanation	ACT	Explanation
HO	H–O	OE	O in 3-membered ring
HN	H–N	OAR	O÷A
HC	H–C	O1S	O=S
HX	other H	OP	O[– =]P
CP	C in 3-membered ring	OX	O[– =]X
C2	C≡A	O1C	O=A
C3O	C=[O S]	O2C	other O
C3N	C=N	S4	O=S=O
C3X	C=A and C–X	S1C	S=C
C3	C=A	SO	S=O
CAR	C÷A	SP	S[– =]P
C4	other C	SX	S=A
NO	N[– ÷ =][O S]	SAR	S÷A
N1	N≡A	S	other S
N2	N=A	P5	five-valency P
NAR	N÷A	P3	other P
N3	other N		

Definitions: –, single bond; =, double bond; ≡, triple bond; ÷, aromatic bond; A, any atom; X, any atom except hydrogen and carbon; [x y z], either of x or y or z.
Scanning is from top to down for each element. For unlisted elements, ACT equals to the element name.

```
C:              ; Carbon type. All after semicolon is ignored.
; carbons in cyclopropane, cyclopropene etc.
                in 3 ring?
                =CP.
; alkyne carbons          ;'-' single bond; '=' double; '#' triple; '~' aromatic
                connected to #*?        ; '*' means any atom.
                =C2.
; (thio)carbonyl carbons
                connected to =[OS]?  ; double bond to O or S, or both
                =C3O.
; imine carbons
                connected to =N?
                =C3N.
; alkene carbons + Halogen or Oxygen
                connected to =* -(?!H\b|C\b)?
                =C3X.    ;(?!..|..|..) – looking for other types than listed ones
; alkene carbons
                connected to =*?
                =C3.
; aromatic carbons
                connected to ~*?
                =CAR.
; alkane carbons + Halogen or Oxygen
;               connected to -(?!H\b|C\b)?
;               =C4X.
; alkane carbons
                =C4.
```

Figure 1. The rule for setting the atom-centered types of a carbon atom.

Fragment generation. The ordinal fragmentation is performed; no fuzzy groups are produced. In fact, discovering all the topological fragments from a molecular graph [15] is a very old problem [16,17]. The common way is the fragmentation of molecular graphs suppressing the isomorphic (duplicated) substructures. However, the solution of the graph isomorphism task on a huge number of fragmental graphs to be generated is quite a time-consuming problem [18]. It can be bypassed by employment of a quickly calculable graph representation as a numeral or a string using them as the hash values of the growing population of fragments (see section '**Fragment ID**' below).

Two new and quite similar algorithms follow another strategy [19,20]. The candidates, as the unlabeled graphs, are grown starting from a single vertex by joining themselves with following testing for their occurrence in molecules. For every molecule, the bit string for presence of previously generated fragments is kept. Therefore, the new larger fragment ("child") can be present in a molecule, if (and only if) both of its "parents" are present, which can be rapidly checked from the bit string. For the positive answer, the subgraph search is executed for confirmation. The authors showed that the exhaustive search is generally very slow; however, an algorithm is suitable for applying the thresholds (*e.g.* the minimal frequency of fragments), and runs much faster as the threshold grows. This type of algorithms deals with the unlabeled fragments only, and therefore is not well applicable for our purposes.

Thus, we will follow the first way of the direct fragmentation of a molecular graph. The task under consideration is formulated as the following: for a given graph, generate all the connected subgraphs, in a way that the number of edges E in any subgraph meets the conditions $E\_min \le E \le E\_max$. Here, E_min and E_max represent the bounds for fragment generating. For $E\_min = 0$, the smallest fragments are just single atoms.

If we "forget" the mentioned need for connectivity, the task will become the classical problem of generating all combinations without repetition of edges within the range E_min ... E_max from N edges of the original molecular graph (we suppose $E\_max \le N$). Then the total number of subgraphs (connected and disconnected) is the sum of binomial coefficients:

$$\sum_{n=E\_\min}^{E\_\max} \frac{N!}{(N-n)!n!}, \qquad (1)$$

An algorithm to generate all combinations without repetition is very simple and intuitive and is not discussed here. This algorithm is not highly effective because of the need of testing for the graph connectivity of appearing fragments. Working around the appearance of disconnected fragments can be reached by another generating algorithm.

For a given molecular graph of N edges (initially the fragment size E is 0), from some starting edge ($E = 1$), its neighbor is selected ($E = 2$). The next edge is also connected to any edge of this conglomerate ($E = 3$). Unless $E = E\_max$, a new edge i ($i \in [1,N]$) can be selected, if it has not been marked to be used before on position $e_i < E$, where e_i was the fragment size E just before the addition of edge i to the grown fragment. Every time the new edge i is added, (a) its e_i is marked and (b) checking for $E\_min \leq E$ is performed and, on success, a new fragment is produced. If there is no edge added, (a) do $E \leftarrow E$-1 and (b) reset all edges' marks e_j ($j \in [1,N]$) greater than E and prune away all those edges from the fragment. When $E = 0$, select another starting edge i, unless $e_i = 0$.

The main difference between the given algorithm and the one for generating all combinations of edges without repetition is the handling with the connected fragment during both growing and pruning stages. An upper bound for the total number of connected fragments is given by equation (1) but the exact estimation is not simple [21]. During generation of fragments, each of them is checked for presence of the user-defined trivial bonds. An example of them is shown in Table 2. A fragment that includes any trivial bond is considered to be trivial on the next steps of our method.

Table 2. The set of trivial molecular bonds used in this analysis.

HC–C4	HC–CP	HC–CAR	HC–C3
HC–C3X	HC–C3N	HC–C2	C4–C4
C4–C3	CAR–CAR	C3–C3	C3=C3

Fragment ID. Several algorithms are known for “compression” of a molecular graph into a single representative value [22]. Among others, the algorithm of Hu and Xu is interesting as simple and quick for computing the numerical identification values [23]. Since the computer manipulation of numbers is always restricted (usually by double precision), the natural weakness of this algorithm is the possibility to get the same hash values for non-isomorphic graphs. However, its authors claimed the algorithm has been tested on a huge dataset of 430472 structures with no collisions. We have adopted this algorithm because the most of the interesting molecular fragments for QSAR are usually short, and there is a small chance for collisions. We introduce the fragment ID based on the molecular identification value obtained by algorithm according to Hu and Xu [23]. The ID value is calculated by manipulation with some form of a path identifier from every atom to all other atoms. The path identifier depends on the atom and bond invariants.

Atomic invariant. Atomic invariant has been suggested as $\delta' = \delta * \sqrt{Z}$, where δ is atom connectivity index as the number of non-hydrogen atoms attached to it; Z is the atomic number. However, for fragments, which are just shivers of a molecule, the counting only the non-hydrogen atoms is not meaningful. Another problem, not covered by Hu and Xu’s work [23], is the lack of support of the single atoms or hydrides like methane; in such cases δ is zero, which is not valid as δ' is used below in the denominator of equation (5). Thus, we assigned δ to the total number of atom’s neighbors; for single atoms, δ is defined to 1/2.

Another particular feature of our fragments is the presence of unions (fuzzy atoms like [C, H] or [S, O, N]) instead of single atoms. The usage of sum of square roots of the shifted atomic numbers (or consecutive AT numbers) among all atoms in the union seems to be a good choice instead of expression $\sqrt{Z}$. Thus, the atom invariant is as follows:

$$\delta' = \delta * \sqrt{Z}, \qquad (2)$$

where δ is degree of vertex (or 1/2 for isolated vertex); and

$$Z = \left[\sum_i \sqrt{\left(\sqrt{2} + Z_i \right)} \right]^2, \tag{3}$$

where square root of two is the irrational shift parameter introduced to prevent possible degenerative cases like this $\sqrt{1} + \sqrt{4} = \sqrt{9}$ when $\sqrt{Z_i}$ becomes integer. Z_i is the atomic number (or AT number) of i^{th} atom from the vertex's union.

Bond invariant. Authors have associated single, double, triple, and aromatic bonds with numbers 1, 2, 3, 1.5, respectively. In this work the fuzzy bonds are also introduced. The bond invariant b is used as the harmonic mean of the particular bond invariants over all corresponding bonds inside the bond's union:

$$b = \frac{n}{\sum_i^n 1/b_i}, \tag{4}$$

where n is the number of bonds in the union, b_i is the bond invariant of each of bond originally in the union ($i \in [1,n]$).

Path identifier. Path identifier between two vertices i and j has been suggested as follows:

$$PI_i = \prod_{k_2}^{n_{ij}} \sqrt{\frac{b_{(k,k-1)}}{k} * \frac{1}{\delta'_k * \delta'_{k-1}}}, \tag{5}$$

where k is the sequence number of vertices along the path between vertices i and j; n_{ij} is the total number of vertices in the path; $b_{(k,k\text{-}1)}$ is the bond invariant between vertices k and k-1; δ' is the atomic invariant. Since root operation is associative ($\sqrt{a} \cdot \sqrt{b} = \sqrt{ab}$), multiplication of square roots will lead to a free mixing of all the data of multipliers. This impediment can easily be avoided utilizing the logarithm function instead of root, like the following:

$$PI_i = \prod_{k_2}^{n_{ij}} \log_e \left(1 + \frac{b_{(k,k-1)}}{k} * \frac{1}{\delta'_k * \delta'_{k-1}} \right). \tag{6}$$

Atomic identifier. Atomic identifier for each atom is calculated by addition of all path identifiers starting from that atom:

$$AID = \sum PI \,. \tag{7}$$

The path identifier *PI* between the same atoms is also used and defined to be 1. Thus, the developed *AID* parameter has the same value 1 for all the smallest fragments – single atoms. Therefore, the following derived value will be the same too and meaningless. Therefore, we used the modified *PI* as following:

$$PI_i = \frac{\sin(3\delta_i')}{5} + \prod_{k_2}^{n_{ij}} \log_e\left(1 + \frac{b_{(k,k-1)}}{k} * \frac{1}{\delta_k' * \delta_{k-1}'}\right). \tag{8}$$

Since the circle's size in radians is irrational, sine seems to produce nearly pseudorandom and unique values for a sequence of possible atomic invariants. The constants of 3 and 5 in equation (8) were selected for convenient scaling only.

Since for the ring systems the multiple inter-atom paths exist, in some complex cases (*e.g.* fullerenes) the counting all possible paths can become an unfeasible task due to its NP-completeness [24]. To simplify the problem, the generation of only the single shortest path (or several paths with the same length) for a pair of atoms was implemented.

Molecular ID. It is a resulting value, which has been defined as the sum of all squared atomic identifiers:

$$MID = \sum AID^2 \,. \tag{9}$$

To prevent the instability of a result due to the rounding errors, we round two last valid digits of *MID*'s mantissa (14^{th} and 15^{th} positions for Intel-style processors) followed by addition of short information about number of atoms and edges in a fragment. For example, the resulting value for union of carbonyl and thiocarbonyl groups C3O=[O1C,S1C] is

represented as “at 2;ed 1;ID 3.553856884348”, which is the fragment ID introduced at the beginning of this section.

For the forthcoming step of fuzzy-fragment creation, the fragments have to be classified according to the topology of their corresponding unlabeled graphs. For this purpose, the fragment ID of the unlabeled skeleton is introduced. It can be calculated easily by setting the atomic number Z and bond invariant b to 1 and calculating the *PI* in accordance to the equation (6).

Fuzzy fragments. At this step, two or more different fragments with the same topology are combined creating a new fuzzy fragment of the same topology with unions of atoms (or bonds) consisting of the corresponding original single atoms or bonds. The fuzziness of atoms or bonds can be defined in advance by the user. The pre-defined accordance set of the available fuzzy matching is presented in Table 3. Notation “N3: C4” means that the atomic type N3 (see Table 1 for notification of AT) from the fragment-pattern can join the atomic type C4 from the fragment-target, but matching *vice versa* is not necessary. Such “unidirectional matching” was introduced to give only the right union because type C4 can not join the type HC or HX, which are incompatible with the type N3. Development of other matching accordance cases was made taking into account such peculiarities. Fuzziness of the different molecular bonds like “Aromatic: Single” can also be defined in our method.

Combination of fragments is done in two steps: for each pattern, (a) try to combine it with every target producing “templates” and (b) make combinations without repetition from the templates found. The fragment-pattern is matched over the fragment-target according to the user-defined fuzzy matching rules. The patterns should only be the non-trivial fragments, but the targets can be either trivial or non-trivial fragments. All fragments to be combined must belong to the same topology.

Templates. Template is the fuzzy fragment where each of fuzzy atom or bond has fewer than 3 components. Combination of pattern and target into a template is the "fuzzy"-graph isomorphism problem. Since no polynomial algorithm is known even for the normal-graph isomorphism testing [25], the problem was formulated as the maximal common "fuzzy" subgraph of two graphs. While the normal (non-fuzzy) problem is NP-complete too (problem GT49 in [24]), we can introduce a quite elegant way for resolving the fuzzy case, which is not so slow in practice.

The maximum common subgraph (MCS) of two graphs is a common subgraph that is not a subgraph of another common subgraph. Its determination can be formulated to search for the clique (the largest complete subgraph) in a modular product [26], which is also known as the docking graph [27]. For two unlabeled graphs U and W, their modular product $U \lozenge W$ is defined on the vertex set $V(U \lozenge W) = V(U) \times V(W)$. An edge between two of its vertices (u_i,w_i) and (u_j,w_j) exists whenever

u_i and u_j in U and w_i and w_j in W are both adjacent: $(u_i,u_j) \in E(U)$ & $(w_i,w_j) \in E(W)$

or

u_i and u_j in U and w_i and w_j in W are both not adjacent: $(u_i,u_j) \notin E(U)$ & $(w_i,w_j) \notin E(W)$

An example of two simple graphs and the corresponding modular product of 9 vertices is presented in Fig. 2. There are several possible cliques available of the cardinality 2, which correspond to all combinations of vertices from matching graphs, except vertices $1'$ and $3'$ from graph W. Those two vertices are the only non-adjacent, and therefore have no correspondence pair in U because all its vertices are adjacent.

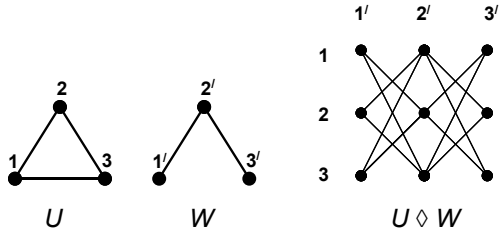


Figure 2. Modular product of unlabeled graphs.

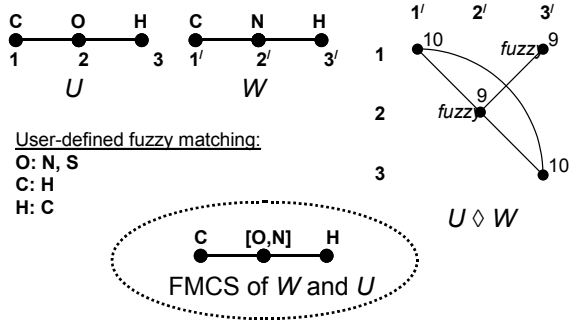


Figure 3. Fuzzy maximum common substructure (FMCS) of molecular graphs W and U.

Table 3. The list of allowable fuzzy matching of atoms and bonds used in this analysis.

HX: HC	OE: CP
HC: HX	OAR: SAR, NAR
	O1C: S1C, N2
C3O: C3N	O2C: S2, N2, N3, C4
C3N: C3O	
C3X: C3, HC, HX	SO: S4
C4: HC, HX	S1C: O1C
CAR: HC, HX	SP: OP
	SX: SO, S4
N1: C2	SAR: OAR
N2: C3, C3X, O1C, S1C	S2: C4, O2C, O2C
NAR: CAR	
N3: C4	CL: BR
	BR: CL

For the case of molecular graphs, graphs U (pattern) and W (target) become labeled by vertices (element names or AT) and edges (type of chemical bond). Therefore, the definition of modular product should be redefined as follows: vertex set $V(U \lozenge W)$ contains of a vertex (u_i,w_i) if and only if vertices u_i and w_i have the same or compatible type. For the latter case, compatibility is defined by the user's fuzzy matching list of atoms (Table 3). Such vertex discrimination significantly decreases the number of nodes in a modular product, if the graphs' vertices differ in their types. In a similar way, the difference in chemical bond types is also used. An edge between two modular product's vertices (u_i,w_i) and (u_j,w_j) exists whenever

u_i and u_j in U and w_i and w_j in W are both adjacent

and with the same or compatible types:

$$(u_i,u_j)\in E(U) \;\&\; (w_i,w_j)\in E(W) \;\&\; type(w_i,w_j)\in type(u_i,u_j)\cup fuzzy\_bonds\_list(type(u_i,u_j))$$

or

u_i and u_j in U and w_i and w_j in W are both not adjacent: $(u_i,u_j)\notin E(U) \;\&\; (w_i,w_j)\notin E(W)$

Obviously, the equality between atom or bond types is more valuable than the fuzzy matching of types. In other words, we have to suppress the appearance of the unneeded inclusion of fuzziness. This will be possible, if we introduce the weights of vertices and edges in the modular product. The vertex weight was defined as 10 units for the exact match of atom types and fewer units (namely 9) for the fuzzy match. The same scheme can be employed for the edges of the modular product.

The concept of the weighted modular product is not known to be introduced elsewhere. For the case of the weighted vertices only (weights of edges are the same), many algorithms exist for detection of clique of the maximal weight [28]. However, for the edge-weighted graph, there is no effective known algorithm and the real challenge is to solve the problems

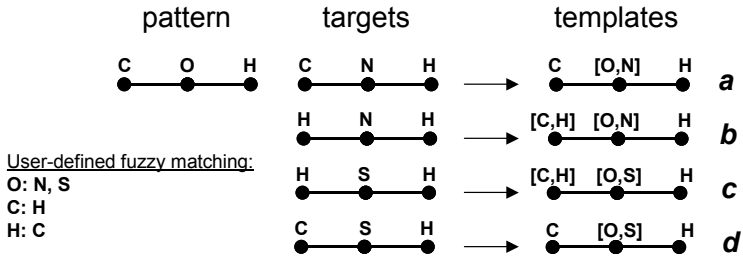
with a number of vertices greater than 40-50 (thus, the maximal cardinality of input graphs for a solvable task is fewer than $\sqrt{50} \approx 7$) [29]. Therefore, we have discarded the weighting the modular product's edges and implementation of priority of the exact bonds over the fuzzy ones. On the other hand, algorithms for detection of the maximum vertex-weight clique are rather fast. We have employed quite effective approach of Östergård, which is capable to handle graphs with as many as thousands of the weighted vertices [30]. Some preliminary modular graph analysis by heuristics is also reasonable to perform check out the feasibility of clique cardinality, which must be equal to the cardinality of input graphs.

The modular product created by these rules is shown in Fig. 3 for two simple fragments of amine and alcohol. According to some user-defined fuzzy matching rules, carbon and hydrogen atoms can not be matched to nitrogen or oxygen atoms, leading to the loss of 4 vertices (1-2′, 2-1′, 2-3′, and 3-2′) in the modular graph. Hydrogen atom can not also be matched over carbon losing vertex 3-1′. Carbon-carbon and hydrogen-hydrogen correspondences are exact and valued as 10 units for such 2 vertices (1-1′ and 3-3′). The other two vertices cost 9 units for the cases of fuzzy matching of the oxygen atom over the nitrogen (2-2′) and carbon over hydrogen (1-3′). The edges of the modular product were set according to the above-discussed rules. The heaviest clique of this modular product has cardinality 3 and weight 29, and includes all three atoms of input graphs. The template produced is a reasonable object from chemical viewpoint.

Combination of fuzzy fragments. They are generated according to the enumeration of combinations without repetition from a set of templates found for a single pattern; the duplicated fragments are suppressed by calculation of the fragment ID. The total number of fuzzy fragments to be generated is estimated according to the equation (1), but many of such fragments may be duplicated. Thus, we have to make all reasonable combinations of 1, 2, …,

n objects. While the parameter *n* is user-definable, we assign it to 3 as a quite reasonable choice; deeper level of fuzziness does not seem to be useful.

Simple example of the process is displayed in Fig. 4. The pattern is a sequence C–O–H, four targets are the fragments of amine, thiol, and a molecule of H_2S. Their combinations, according to the upper section, produced four templates ***a***, ***b***, ***c***, and ***d***. The combining of the templates gave us 4+6+3 fuzzy fragments with different fuzziness level, and some of them were duplicated. The final output consisted of 6 fuzzy fragments only. We called them the molecular "Fuzzy Characteristic Groups" (FCG) due to the absence of trivial bonds, as their parent pattern consisted of the non-trivial bonds exclusively.



Combinations of level 1: ***a***, ***b***, ***c***, ***d***

Combinations of level 2: ***ab*** (equal to ***b***), ***ac*** ([C,H]-[O,N,S]-H), ***ad*** (C-[O,N,S]-H), ***bc*** (equal to ***ac***), ***bd*** (equal to ***ac***), ***cd*** (equal to ***c***)

Combinations of level 3: ***abc*** (equal to ***ac***), ***abd*** (equal to ***ac***), ***bcd*** (equal to ***ac***)

Output fuzzy fragments: ***a, b, c, d, ac, ad***

Figure 4. Producing fuzzy fragments.

Fragment counting. Many learning methods for the data mining in QSAR analysis, *e.g.* the rule-based approaches, need only the unary categorical descriptors as an input (present/absent). Therefore, in such cases the step of computing the frequency of fragments in a molecule may be omitted. It is particularly important, because the counting fragments may be generally considered as the most time-consuming of the whole approach, because of the presence of possibly large substructural search tasks known to be NP-complete.

We can evaluate the complexity of the substructure search task. For example, a typical molecule may have 40 atoms, while a fragment may have 10 atoms. Therefore, the maximal cardinality of the modular product is up to 400 vertices. It is quite a large graph and the detection of its clique may be very hard in the worst case. Of course, there may be larger molecules and fragments as well.

We have tried to decrease the computational cost by a special technique handling the edge graphs [26,31]. Nevertheless, the problem remains to be NP-complete and, if a target molecule and a pattern are large structures and have many degenerative bonds, the search for the solution may also be very hard, unless an approximation algorithm is preferred.

In the case of molecular graph, the discrimination between two vertices (atoms) is much smaller than between two edges including the endpoints (chemical bonds). It is because the bond contains the information about the types of two atoms and the type of bond. All this information must be in conformity between the two bonds (of both target and pattern structures) being matched. This way allows us to dramatically decrease the modular-product size when the structures do not contain a lot of degenerative bonds. On the other hand, the edge density of such modular product often becomes lower as well. All this facilitates the clique detection.

An edge graph associated with graph $G = [V,E]$ is denoted by $Edge(G) = [E,I]$, if every vertex $v \in Edge(G)$ is edge $e \in E$; two of these vertices in $Edge(G)$ are neighbors, if the two

corresponding edges in *G* have a common endpoint (see Fig. 5). However, this mapping is not always isomorphic: there are exactly five graphs where edge graph isomorphism is not fulfilled [31,32]. Those are presented in Fig. 6. Ambiguous edge graphs are shown for cases **a**, **b**, and **c**. The other two cases **d** and **e** are similar to graph **c**, but have more edge permutations.

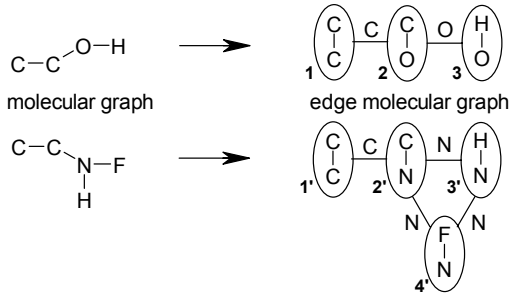


Figure 5. Examples of edge molecular graphs.

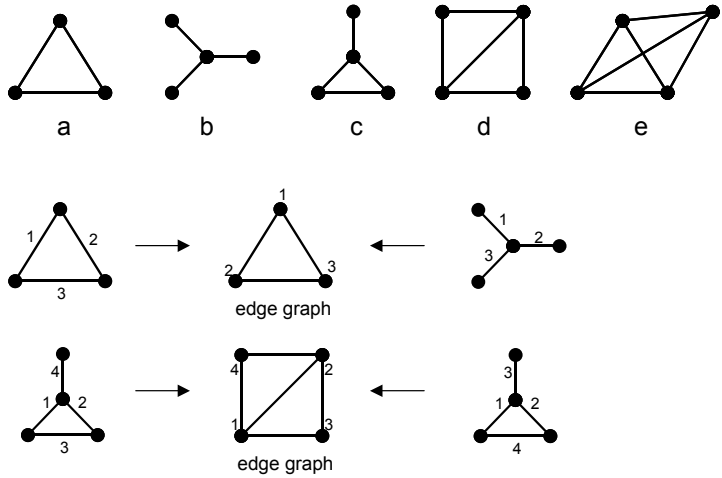


Figure 6. Enumeration of graphs (a-e) that do not have one-to-one edge isomorphism transformation.

Clearly, chemical structures with the topology given by graphs **d** and **e** are rather exotic and unstable because of great bond and angle tensions. In any case, they are completely uncommon in the real practice and can be neglected.

On the other hand, structures **a**, **b**, and **c** are very common, and must be handled. It can be done by two different ways. The first one is just the catch of the corresponding input graphs with the topology of **b** (or **a**) and **c** followed by a special processing. The second one is the detection of 3-membered rings and marking bonds of such rings as the specific ones [26]. Such marked bonds differ from the other bonds; therefore, the edge isomorphism ambiguity can not arise. This way is also chemically reasonable because of the fact that the bonds in 3-membered ring have quite specific behavior, and are often called as the “banana” bonds [33]. In this work, the second way was applied.

Once the edge graphs are formed from the initial molecular graphs of a pattern (a molecular fragment, possibly fuzzy) and a target (a molecule from a data set), the creation of modular product followed by the clique search is processed by the same way as described in the previous section. The difference is in absence of weighted vertices of modular graph and in introducing some heuristics optimizing the substructure search.

The example of the substructure search in the edge graph is shown in Fig. 7. The fragment is hydrogen-depleted ethylene oxide; the target is its tetramethyl derivative (no hydrogen atoms). *Edge(U)* and *Edge(W)* are their edge graphs, respectively. In the lower part of the picture, the modular product of molecular graphs (left) and their edge graphs (right) is presented. We can notice a few things while observing this figure. First, the cardinalities of the edge graphs are the same with cardinalities of the original graphs in both cases. This principle is quite common for real molecular graphs, because of the low order of vertex degree due to the valency restrictions: the number of bonds in molecules is normally close to the number of atoms.

Second, the cardinality of a modular product was dramatically decreased for the case of edge graphs. This was reached due to the difference between the outer carbon-carbon bond and that in the 3-member ring. In other cases, such a difference may be caused by the distinction in the type of bond or anyone of endpoints (atoms). Such simplification has been the main goal of the usage of the edge graph concept for substructure search. Since the clique detection is the NP-complete task, decrease of the size of its input graph greatly accelerates the whole process. The clique size for both modular products from Fig. 7 is three, but the clique detection for the case of the edge graphs is undoubtedly easier.

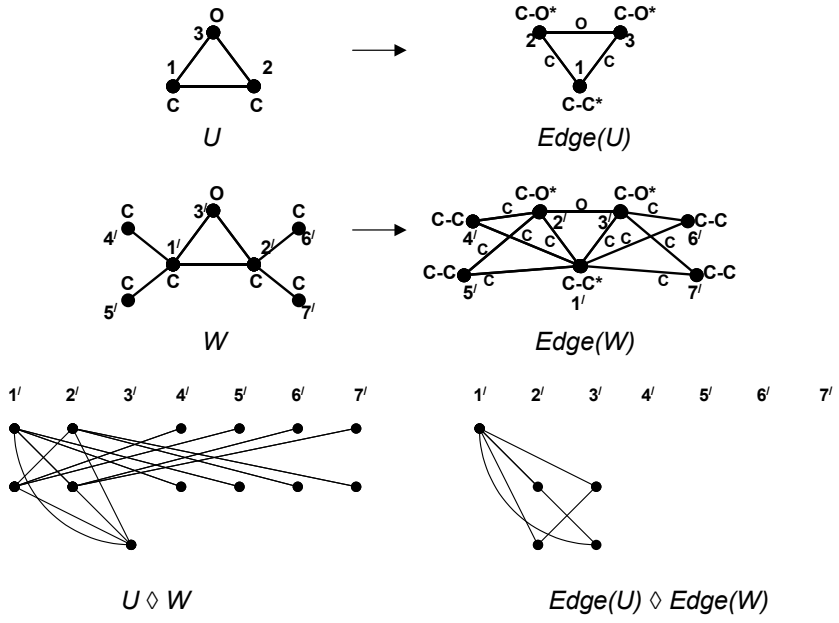


Figure 7. Substructure search in molecular graphs and corresponding edge graphs.

QSAR Experiment

The QSAR problem for an assessment of the applicability of the introduced FCG descriptors was the explanation of 96h acute toxicity (expressed as negative logarithm of 50% lethal concentration) against *fathead minnow* [34,35]. The data set contains 478 diverse organic molecules of 3 toxicity classes as non-polar narcotics, polar narcotics, and reactive toxicants. Such separation is qualitative and variable in many cases. The toxicity data had been analyzed by the authors using the artificial neural network modeling with 16 structural descriptors and calculated logarithm of *n*-octanol-water partition coefficient [35]. The whole data set was split onto a training set of 384 molecules and a test set of 94 molecules. The neural network analysis gave us the results with correlation coefficient $R = 0.819$ and root mean-squared error $RMS = 0.676$ for the training set, and for the test set, $R = 0.737$ and $RMS = 0.811$. Obviously, one of the reasons for such a low quality of the regression is due to the presence of too many different animal toxicity mechanisms [36].

In the present work, we used the same training and test sets. The molecular structures were prepared as the connection tables. The aromaticity of rings was checked by the Hueckel rule.

For each molecule, the atom types were assigned according to Table 1. All bonds between the hydrogen and sp^2 or sp^3 hybridized carbon atoms were removed from the molecular structures as redundant; such bonds are namely HC-C4, HC-CAR, and HC-C3. The lists of trivial bonds and allowable fuzzy matching of atoms were shown in Tables 2 and 3, respectively. The lists of AT and fuzzy matching are quite versatile because they cover various aspects of similarity and diversity between different atoms. The need of usage of such numerous multi-parametric options was due to the well-known complexity of the toxic action as a molecule may have a wide variety of targets in the living organism.

The summary of generation of fragments up to 7 bonds is presented in Table 4. For Partial Least Squares (PLS) analysis, fragments of size 0, 1, and 2 only were selected, and their

support as the number of compounds with a given fragment was calculated. Thus, in total there were 1152 unique trivial, non-trivial, and fuzzy fragments. Seventy five trivial fragments and 183 fragments with the low support (≤ 2) were excluded. As the next step, the degenerative and near degenerative cases were suppressed. For each pair of fragments with less than 3 distinctions in the number of occurrences or with the correlation coefficient greater than 0.95, the fragment with larger number of bonds was excluded. Finally, only 308 descriptors were retained.

PLS analysis was carried out on the training set of 384 molecules with all 308 FCG descriptors, and validated on the test set. Thirteen PLS components were extracted to get the lowest error on the test set resulting in the following modeling quality: for the training set, R = 0.871, RMS = 0.559; for the test set, R = 0.838, RMS = 0.595.

The found PLS model utilizes only the suggested FCG molecular descriptors, thus achieving both higher quality of the model and lower prediction error, especially for the test set. No tuning to minimize the error on the test set has been done, as opposing to the neural network modeling [35]. The correlation chart for QSAR modeling of the fish toxicity is shown in Fig. 8. Most outliers are located under the regression line and have high observed toxicity. There are two reasons for such underestimation of toxicity values. Several compounds have rare but highly reactive combinations of functional groups, for example, 2,2,2-trifluoroethanol, 2-chloroethanol, and chloroacetonitrile. Some molecules may be called “singletons” as they bear certain unique structural features, which do not appear in other molecular structures from the data set. Such low-support fragments have been naturally suppressed due to their low statistical significance. Thus, the fragments with possible great contribution were not taken into consideration. Iodine atoms in 3,5-diiodo-4-hydroxybenzonitrile and 2,4,6-triiodophenol are such examples. On the other hand, a few chemicals of low observed toxicity had slightly overestimated predicted values:

saccharin, 5,5-dimethyl-1,3-cyclohexanedione, and ethyl trifluoroacetate. In spite of the visually harmful structural features, saccharin (*o*-sulfobenzoic acid imide) is well-known to have a unique low toxicity being the world's oldest artificial sweetener. Reasons of the low toxicity for the latter compounds are not evident.

It was demonstrated by the Table 4 that the total number of fragments generated for the diverse data set becomes very large. The table shows the following points:

1. The number of trivial fragments increases by a nearly geometric law of fragmental size (the factor is a bit more than two).
2. The number of non-trivial fragments increases only slightly, but the number of fuzzy fragments goes over the maximum.
3. The fragment support decreases when the fragment size increases, especially for the exact fragments. Fragments with the low support produce only near-constant descriptors, which are useless for any QSAR analysis.
4. The support of fuzzy fragments is much greater than that of the exact fragments. This peculiarity provides a way to keep information about the rare molecular features. The low support fragments themselves have no statistical significance.
5. For the given data set, computation time is quite acceptable. For the smallest fragments, the computation time for the counting stage was much shorter because of the implementation of specific heuristics. For other cases, the slower subgraph search by clique detection in a modular product was performed. The counting of size-3 fragments was slow due to the frequent occurrences of certain fragments in the molecules.

We demonstrated that the large descriptor space could be efficiently cut down removing the near-constant and near-degenerative cases. General data reduction techniques as partial linear regression also provide the powerful way for the further data compression.

Table 4. Summary of fragment generation for 478 molecules.

Fragmental size	Exact fragments trivial [a)]	non-trivial [a)]	time (s)	Fuzzy fragments quantity [a)]	time (s)	Counting time (s)
0	- / -	63.7 / 38	6.8	254.9 / 56	0.3	8.3
1	66.6 / 9	19.2 / 83	3.7	120.2 / 218	4.3	23.5
2	21.0 / 66	7.9 / 175	5.8	63.2 / 507	14.4	41.5
3	9.6 / 315	3.3 / 285	9.7	34.8 / 711	28.2	782.2
4	5.2 / 952	1.9 / 357	17.7	15.6 / 705	58.2	452.5
5	3.4 / 2325	1.5 / 378	32.9	8.3 / 469	72.8	357.4
6	2.5 / 4804	1.5 / 388	59.2	5.4 / 257	112.6	411.0
7	1.9 / 9094	1.5 / 393	108.9	4.5 / 160	114.3	594.7

a) – average support / total number of fragments.

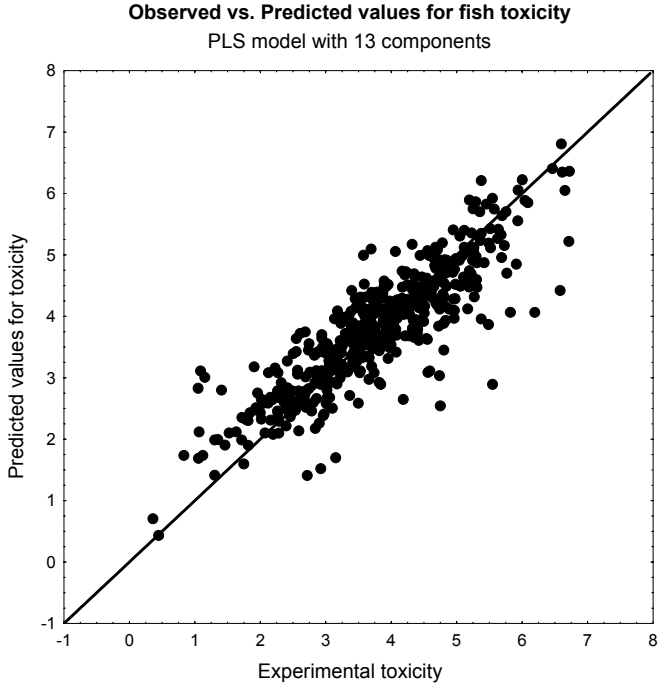


Figure 8. Scatter plot for toxicity of 478 molecules.

Conclusion

The Fuzzy Characteristic Groups (FCG) have been proposed as the new topological molecular descriptors. The generation of descriptors can be tuned widely according to the user's idea about a given QSAR problem. The physical meaning of the FCG is easily understandable, unlike many other molecular descriptors used for QSAR. This fact provides the realistic way for developing clear relationships between molecular properties and structure. The developed descriptors have been successfully applied to a QSAR problem of toxicity of the large heterogeneous data set of toxicants with different mechanisms of acting.

Acknowledgement

This work was partially supported by Japan Chemical Industry Association.

References

1. T.I. Oprea, C.L. Waller, Theoretical and Practical aspects of Three-Dimentional Quantitative Structure-Activity Relationship. In: K.B. Lipkowitz, and D.B. Boyd (Eds.), Reviews in computational chemistry, VCH Publishers, New York, 11 (1997) 127-182.
2. I.H. Witten, E. Frank, Data mining. Academic Press, San Diego, 2000.
3. M. Karelson, Molecular descriptors in QSAR/QSPR, Wiley-Interscience, New York, 2000.
4. M.T.D. Cronin, T.W. Schultz, Pitfalls in QSAR, J. Mol. Struct. (Theochem) 622 (2003) 39-51.
5. J. Devillers, Environmental Chemistry: QSAR, in "Encyclopedia of Computational Chemistry," Wiley, Chichester (1998), pp. 932-941.
6. P.A. Carrupt, B. Testa, P. Gaillard, Computational approaches to lipophilicity: methods and applications. In: K.B. Lipkowitz, and D.B. Boyd (Eds.), Reviews in computational chemistry, VCH Publishers, New York, 11 (1997) 241-315.
7. G. Klopman, H. Zhu, Estimation of the Aqueous Solubility of Organic Molecules by the Group Contribution Approach. J. Chem. Inf. Comput. Sci. 41 (2001) 439-455.
8. J.W. Raymond, T.N. Rogers, D.R. Shonnard, A.A. Kline, A review of structure-based biodegradation estimation methods. J. Hazard. Mater. 84 (2001) 189-215.
9. H.J.M. Verhaar, E.U. Ramos, J.L.M. Hermens, Classifying environmental pollutants 2: seperation of class 1 (baseline toxicity) and class 2 ('polar narcosis') type compounds based on chemical descriptors. J. Chemometr. 10 (1996) 149-162.

10. A.R. Katritzky, D.B. Tatham, U. Maran, Theoretical Descriptors for the Correlation of Aquatic Toxicity of Environmental Pollutants by Quantitative Structure Toxicity Relationships. J. Chem. Inf. Comput. Sci. 41 (2001) 1162-1176.
11. N.S. Zefirov, V.A. Palyulin, Fragmental approach in QSAR. J. Chem. Inf. Comput. Sci. 42 (2002) 1112-1122.
12. V.N. Viswanadhan, A.K. Ghose, G.R. Revankar, R.K. Robins, Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive iteractions and their application for an automated superposition of certain naturally occuring nucleoside antibiotics. J. Chem. Inf. Comput. Sci. 29 (1989) 163-172.
13. G. Klopman, J.Y. Li, Sh. Wang, M. Dimayuga, Computed log P Calculations Based on an Extended Group Contribution Approach. J. Chem. Inf. Comput. Sci. 34 (1994) 752-781.
14. L. Wall, T. Christiansen, J. Orward, Programming Perl. O'Reilly, Beijing, 2000.
15. D.H. Rouvray, A.T. Balaban, Chemical applications of graph theory. In: R.J. Wilson, and L.W. Beineke (Eds.), Applications of graph theory, Academic Press, London, (1979) 177-221.
16. Y. Takahashi, Y, Satoh, H. Suzuki, H. Abe, Sh. Sasaki, Enumeration of unique substructures from a chemical structure. Anal. Sci. 2 (1986) 321-323.
17. V.N. Piottukh-Peletsky, A.K. Rumyantsev, V.I. Smirnov, Application of the new chemical structure representation based on the complete set of fragments to fast substructure search and classification, Proc. of the 4th Japan-USSR symposium on computer chemistry, Toyohashi, Japan (1991) 8-9.
18. B.D. McKay, Practical graph isomorphism. Congr. Numerantium 30 (1981) 45-87. For software see also URL: http://cs.anu.edu.au/people/bdm/nauty/.
19. A. Inokuchi, T. Washio, Y. Nishimura, H. Motoda, General framework for mining frequent patterns in structures, Proc. of international workshop on active mining (IEEE ICDM), Maebashi, Japan (2002) 23-30.
20. M. Kuramochi, G. Karypis, An efficient algorithm for discovering frequent subgraphs, Technical Report 02-026, Department of Computer Science, University of Minnesota, 2002. See also URL: http://www.cs.umn.edu/~kuram/papers/fsg-long.pdf.
21. Rücker G., Rücker Ch.: Automatic Enumeration of All Connected Subgraphs MATCH Commun. Math. Comput. Chem. 41 (2000), 145-149.
22. O. Ivanciuc, Canonical numbering and constitutional symmetry. In: R. Schleyer (Ed. in Chief), Encyclopedia of computational chemistry, Wiley, Chichester (1998) 167-183.
23. C.Y. Hu, L. Xu, Developing Molecular Identification Numbers by an All-Path Method, J. Chem. Inf. Comput. Sci. 37 (1997) 311-315.
24. M.R. Garey, D.S. Johnson, Computers and intractability – a guide to the theory of NP-completeness. Freeman, San Francisco, 1979.

25. S. Skiena, Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Addison-Wesley, Redwood, 1990.
26. J.W. Raymond, E.J. Gardiner, P. Willett, Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm, J. Chem. Inf. Comput. Sci. 42 (2002) 305-316.
27. Y. Takahashi, Identification of structural similarity of organic molecules. In: K. Sen (Ed.), Molecular similarity II, Springer, Berlin, (1995) 105-133.
28. I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem. In: D.-Z. Du and P.M. Pardalos (Eds.), Handbook of combinatorial optimization, Supplement volume A, Kluwer, Dordrecht, (1999) 1-74.
29. U. Faugle, G. Dijkhuizen, A cutting-plane approach to the edge-weighted maximal clique problem, Eur. J. Oper. Res. 69 (1993) 121-130.
30. P.R.J. Östergård, A new algorithm for the maximum-weight clique problem, Nordic J. Comput. 8 (2001) 424-436.
31. V. Nicholson, C.-C. Tsai, M. Johnson, M. Naim, A subgraph isomorphism theorem for molecular graph. In: R.B. King, and D.H. Rouvray (Eds.), Graph theory and topology in chemistry, Elsevier, Amsterdam (1987) 226-230.
32. M. Behzad, G. Chartrand, L. Lesniak-Foster, Graphs and digraphs. Wadsworth, Belmond, 1979.
33. A.D. Walsh, Structures of ethylene oxide and cyclopropane, Nature (London), 159 (1947) 712-713.
34. M. Nendza and C.L. Russom, QSAR modeling of the ERL-D *fathead minnow* acute toxicity database, Xenobiotica, 21 (1991) 147-170.
35. G. Admans, Y. Takahashi, S. Ban, H. Kato, H. Abe, S. Hanai, Artificial neural network for predicting the toxicity of organic molecules, Bull. Chem. Soc. Jpn., 74 (2001) 2451-2461.
36. T.R. Stouch, J.R. Kenyon, S.R. Johnson, X.-Q. Chen, A. Doweyko, Y. Li, In silico ADME/TOX: Why Models Fail, J. Comput.-Aided Mol. Des., 17 (2003) 83-92.