Sidorova, J., García, J. (2015). Bridging from syntactic to statistical methods: Classification with automatically segmented features from sequences. *Pattern Recognition*, 48, pp. 3749-3756.

DOI: 10.1016/j.patcog.2015.05.001

Bridging from Syntactic to Statistical Methods: Classification with Automatically Segmented Features from Sequences

J. Sidorova[a,b], J. Garcia[a]

[a] *Lab for Applied Artificial Intelligence, Faculty of Computer Science, Universidad Carlos III de Madrid.*
[b] *Department of Computer Science and Engineering Blekinge Institute of Technology, Sweden.*
julia.a.sidorova@gmail.com

Abstract

To integrate the benefits of statistical methods into syntactic pattern recognition, a *Bridging Approach* is proposed: (i) acquisition of a grammar per recognition class; (ii) comparison of the obtained grammars in order to find substructures of interest represented as sequences of terminal and/or non-terminal symbols and filling the feature vector with their counts; (iii) hierarchical feature selection and hierarchical classification, deducing and accounting for the domain taxonomy. The bridging approach has the benefits of syntactic methods: preserves structural relations and gives insights into the problem. Yet, it does not imply distance calculations and, thus, saves a non-trivial task-dependent design step. Instead it relies on statistical classification from many features. Our experiments concern a difficult problem of chemical toxicity prediction. The code and the data set are open-source.

Keywords: syntactic pattern recognition, grammatical inference, feature

April 30, 2015

segmentation, SMILES parser, feature extraction,

1. Introduction

Statistical pattern recognition has a simple representation in the form of vectors allowing efficient ways to manipulate them, while syntactic pattern recognition has expressive representations, – graphs, strings, and so on, – but lacks object manipulation tools. Until recently, the syntactic and structural communities coexisted without much interaction. Yet, with the ever increasing difficulty of tasks in pattern recognition, more and more often the questions are asked: *–Can we have advantages of both paradigms? –Which are the trade-offs in such combinations?*

Syntactic pattern recognition can be used if there is a clear structure in the patterns and a *grammar* can be observed in a natural way. Forcing modeling on data, e.g. imposing linear ordering, hampers the performance [1]. Objects are represented by a variable-cardinality set of symbolic features.

Let there be n different grammars $G_1, ..., G_n$, one for each recognition class C_k $k = 1, .., n$. A *pattern* p_x of an object x, – where, x can be a written digit, speech sample, protein sequence, etc. – must first be transformed to a sequence of terminal symbols, that is, smallest units. For example, a protein sequence as a string

$$p_x = ATTTGGGGCTTATATAT, \tag{1}$$

where A, T, C, G are terminal symbols corresponding to the four nucleotides in the DNA. Examples of a recognition class C_k form a training set $S(C_k)$:

$$S(C_k) = \{p_{k_1}, p_{k_2}, p_{k_3}, ...\}, \tag{2}$$

and a grammar G_k is sought, such that $L(G_k) \supseteq S(C_k)$. For a review of grammatical inference issues the reader is referred to [2], [3].

There exist various *distance metrics* to measure similarity between patterns. Let $D(p_x, C_k)$ be some distance from a pattern p_x to a class C_k. The (smallest) distance between an input pattern p_x and a recognition class C_k[1] is

$$D(p_x, C_k) = min\{D(p_x, p_k) | p_k \in L(G_k)\}. \quad (3)$$

In the literature, *three main approaches* to syntactic pattern recognition are typically singled out [4]:

– with an error-correcting parser,

– distance-based, and

– stochastic.

An error-correcting parser decides whether p_x belongs to $L(G_i)$ or not. If p_x belongs to $L(G_i)$, x is assigned to category C_i, and it is rejected otherwise. *The distance-based scheme* computes a distance from p_x to $L(G_k)$. If $D(p_x, L(G_i))$ is smallest among all the classes $C_1 ... C_n$, x is assigned to category C_i. Here, a statistical component is often added, and the distances to recognition classes are the input to a statistical classifier, where C4.5 or the kNN are known to perform well and keep the classification process human readable. *Stochastic schemes* consist in adding occurrence probabilities to productions in the schemes defined above.

Obviously, object representation is crucial, and *graphs* would be ideal in many applications, but learning graph grammars is largely infeasible due to

[1] or equivalently, between p_x and $L(G_k)$

complexity issues[2], instead graph embedding, e.g. [6], [7], and kernel methods, e.g. [8], [9], are used. For the research trend on graphs in pattern recognition, the reader is referred to [10]. *Strings* are suitable, since a regular or context-free grammar can be efficiently learnt and similarity measures calculated. If the target language is regular, hidden Markov models (HMMs) have been used in many applications [11]. For example, they are the mainstream tool to discover chromatin states [12], or protein regions [13] with distinct biological functions. The problem is that HMMs treat sequences as one-dimensional strings of independent, uncorrelated symbols. Although computationally convenient, this assumption is not structurally realistic [14], because many phenomena have more complex structure than regular: natural language, palindrome structures in biology, and so on. Furthermore, once the target structure rises in terms of structural complexity from regular to context-free, one must make quite a number of task-dependent modeling decisions, and as a result applications become harder to design and reuse. Still, such efforts exist in optical character recognition [15], analysis of coronary artery images [16], in chemical biodegradability prediction [17], [18], and some other.

Statistical pattern recognition has a simple representation in the form of vectors and efficient ways to manipulate them [19]. It has gained a much greater popularity than the syntactic paradigm. Yet, faced with ever growing difficulty of tasks, a recent tendency is to adapt ideas from syntactic

[2]A problem of parsing non-trivial graph languages is PSPACE-complete or NP-complete. Defining graph-grammars generating languages with a polynomial membership problem is an open problem [5].

methods. For example, in image understanding, ontologies are used for the loss function design: it is less of an error to take a cat for a dog, since both are animals, than a cat for a truck. In image tagging, structurally related features were shown to improve performance: if a ship has been detected, the probability for the the sea should be high. In graph matching, structural information allows for constraint formulation: if a face is adjacent to a neck in one graph, it should be so in the other one, too. For an overview the reader is referred to [20]. Another idea proposed is to gain interpretability of predictive models in some creative task-dependent way, which often comes with a cost in recognition accuracy compared to black-box solutions or may require that the underlying linear model works well on the data set: for example, adding a heat map coloring technique to interprete linear support vector machine models [21].

This work, too, explores connections between the two paradigms, but our idea is different. In our previous work [18], we departed from the fact that there is a grammar for chemicals, very much like a natural grammar, and, we designed a syntactic pattern recognition scheme together with a procedure to search for important substructures in the grammars. In this submission, we propose to fill the feature vector with the counts of potentially important substructures. These substructures are automatically segmented, have an automatically chosen degree of structural abstraction and special statistical properties. The proposed *Bridging Approach* brings the following benefits:

1. The method's essential capacity is *to cope in the absence of expert knowledge*, that is, no indications with respect to which features to extract or where to look for them in the input sequence.

2. *It gives insights into the problem* in two respects. Firstly, the method works with a variable-length parsable input and finds the regions of interest in sequences with a suitable level of abstraction for their representation. Secondly, subsequent hierarchical vector-based feature selection and classification account for the domain's taxonomy.
3. *It is easier-to-implement* than a classical syntactic scheme, since it does not imply distance calculations. Therefore, it saves a non-trivial design step from the syntactic paradigm.

Our experiments concern a difficult problem of chemical toxicity prediction. Our parser processes molecules in the SMILES format, which is a string representation of a 2D molecular graph. From two sets of molecules with opposite properties $S(G_{\oplus})$ and $S(G_{\ominus})$, a predictive model is built with *the Bridging Approach*.

The rest of the paper is organized as follows. Section 2 explains how chemicals are represented as strings and how they are parsed. Section 3 explains the steps of *the Bridging Approach*. Section 4 covers the experiment. Finally, conclusions are drawn in Section 5. The SMILES parser and *the bridging approach* are available on request from the corresponding author. The database used for experiments is NCTRER DSSTOX at hppt://www.epa.gov/nheerl/dsstox/sdf_nctrer.html

2. Parsing Chemicals

The chemical language SMILES was designed "*to represent molecular structure by a linear string of symbols, similar to a natural language*" [22]. A sequence in SMILES represents a molecular structure as a graph.

Atoms: Atoms are represented by their atomic symbols: C, Cl, N, O, etc. This is the only required use of letters in SMILES. Hydrogen atoms (H) are normally omitted, since valences make it clear where they are missing. For example, an atomic chain CCSCCCCC[3] is depicted in Figure 1.

Bonds: Single bonds are usually omitted in SMILES. Double and triple bonds are represented by the symbols = and #, respectively, for example, in Figure 2.

Branches. Branches are specified by enclosures in parentheses, as in Figure 3.

Cyclic Structures: Cyclic structures are represented by breaking one single (or aromatic) bond in each ring. The bonds are numbered in any order, designating ring opening (or ring-closure) bonds by a digit immediately following the atomic symbol at each ring closure. This leaves a connected noncyclic graph, which is written as a noncyclic structure, as in Figure 4.

With the rules above almost all organic structures can be described as strings. For more details, the reader is referred to [22].

A context-free parser based on the SMILES grammar we developed creates a syntax tree from SMILES, see Appendix A for further details.

3. The Bridging Approach

Input is parsed structured data, the *Bridging Approach* will study it and build a predictive model based on its conclusions. Briefly, its steps are:

1. acquisition of a grammar per recognition class;

[3]Due to chemical convention in graphics, whenever a label on graph node is missing, it is C and a line segment represents a chemical bond.

2. comparison of the obtained grammars in order to find substructures of interest represented as sequences of terminal and/or non-terminal symbols and filling the feature vector with their counts;
3. hierarchical feature selection and hierarchical classification, deducing and accounting for the domain taxonomy.

Step 1: acquisition of a grammar per recognition class.

The general assumption is that objects with similar structures have similar properties. Given two sets of examples from opposite classes (for example, active and non-active chemicals), we can learn grammars that account for their structures: $L(G_k) \supseteq S(C_k)$. Examples are taken from the training set one by one. Whenever an example cannot be parsed with the current grammar, the grammar is extended with new rules to accommodate the example.

The grammar inference algorithm from SMILES [18] is reproduced in Appendix B. The input to it is a training set with parsed SMILES of chemicals belonging to the same activity class, and the output is the grammar G and table T of two columns:

$$\langle \text{production } p \rangle \text{ and } \langle \text{how many times } p \text{ was used} \rangle.$$

Step 2: comparison of the obtained grammars in order to reveal substructures of interest represented as sequences of terminal and/or non-terminal symbols.

For a binary problem[4], the tables for the two classes, $T_{\oplus}$ and $T_{\ominus}$, are compared, in order to search for the substructures of interest. There is a

[4]A multiclass problem can be recast into a series of binary classification problems with one-versus-all [23], one-versus-one [24] and error-correcting output codes [25], [26].

qualitative and quantitative aspect to this search. The qualitative aspect concerns feature segmentation: what are the substructures of interest and how they are represented as terminal and/or non-terminal symbols. The quantitative question is which statistical properties the substructures should possess, in order that the counts of their occurrences can be useful as features in statistical classification.

The qualitative issue is resolved by the grammar. Consider examples of productions:

$$sig_2 \rightarrow sig_6 sig_6, \tag{4}$$

$$sig_6 \rightarrow C1Csig_3CCC1, \tag{5}$$

$$sig_6 \rightarrow C1CCCCC1. \tag{6}$$

The left-hand side of productions entirely depends on the right hand side and is redundant, that is, it is of the form sig_{arity}, where the arity is the number of units that appear in the right-hand side. Thus, we can work with the right-hand side only. The grammar defines how the substructures are segmented and the level of abstraction. In our example the substructures are $sig_6 sig_6$, C1Csig_3CCC1, C1CCCCC1.

From a quantitative perspective, naturally one would look closely at frequently encountered molecular substructures that are *exclusive* for one class. Unfortunately, such ideal "structural alerts" are infrequent due to many chemical exceptions, and we can't hope that they alone can solve the classification and explanatory tasks. *Common substructures* need to be considered. In order to favor the ones that are more frequent in one class and less frequent in the other, the ones that have the importance value greater than average

are taken, where

$$Importance(X_i) = |count(X_i)\ in\ T_{\ominus} - count(X_i)\ in\ T_{\oplus}|. \quad (7)$$

Step 3: hierarchical feature selection from the pool of substructures of interest and hierarchical classification, deducing and accounting for the domain taxonomy at the feature selection and classification steps.

Initially, two types of substructures are filtered: the substructures found in one of the classes exclusively and common substructures that are more frequent to in class than in the other. In order to incorporate this intuition into a predictive model, it needs to be backed with statistics. Additionally:

1. It should be taken into account that many domains have natural taxonomies, for example species, chemicals etc form families and subfamilies. Within a taxonomic category, objects have comparable structures and property-specific structural clues can further be discovered. In terms of structure (morphology), the gold fish can be compared to the carp, but not to the hamster.
2. The method should keep the criteria for classification human-readable.

Decision trees are a standard choice, when human readability and gaining insights are sought. Further, C4.5 [27] automatically partitions the feature space and chooses appropriate features for classification in each subregion.

4. Experiment

Data: A large number of chemicals present in the environment are estrogens, that is they are structurally similar to hormones and disrupt en-

docrine functions in animals and humans [28]. The NCTR (National Center for Toxicological Research) Estrogen Receptor Binding database [29], hppt://www.epa.gov/nheerl/dsstox/sdf_nctrer.html, consists of 232 chemicals. Its creation was motivated by the desire to summarize the knowledge about estrogens and have a reliable data set of consistent design that would fully cover structurally diverse set of natural, synthetic, and environmental estrogens. Once the list of chemicals had been composed by experts, they were tested on rats in well validated and standardized analytical procedure. The estrogen activity was measured on the scale from 0 to 100: 0 corresponds to *inactive*, the chemicals with the activity values ≥ 23 are labeled as *active*, and the structures labeled as *inconclusive* have the activity value equal to 5. The authors also provided a set of chemical rules linking substructures, types of chemicals and their resulting activity. Among the 232 samples: 89 chemicals are active (the $\oplus$ class), and 123 are inactive (the $\ominus$ class), and 8 chemicals are labeled as *inconclusive*. We decided to include the inconclusive chemicals as a third class to observe the tendencies.

Learning settings: the experiments were carried out in 10-fold cross validation. A 90% part of data was taken for training purposes to carry out steps 1-3 of the *Bridging Approach*. The remaining 10% was used for testing.

Results: The confusion matrix for the experiment is presented as Table 1. As had been expected, the grammar for the inconclusive class was very small (since it had too few training examples) and therefore useless. Consequentially, the inconclusive class could not be recognized, and its chemicals appeared to fall randomly into the active and inactive classes. Further, when calculating recognition accuracy and other characteristics, the inconclusive

class was not taken into account. The overall recognition is 75%, recall = 0.69, precision = 0.71 and F-measure = 0.7.

A predictive model is considered successful, if its accuracy is better than 70% [30]. Our result is above this baseline, is comparable to some studies on the same database (67% [31], 78% [32], 79% [33]), but is considerably behind the best result reported of 85% [34] with two black-box methods that implemented the Random Forest with 500 trees and Classification by Ensembles from Random Partitions (CERP). That is in line with the literature, e.g. [34]: readability often goes with a cost in accuracy.

Once the recognition capacity of the method had been concluded to be satisfactory, the learning procedure was repeated on the whole dataset to obtain a decision tree that summarizes the activity in terms of structural features, which is depicted in Figure 12 in Appendix C.

The database creators provided an expert model with *if- then-* rules to summarize extrogenic activity based on advanced expertise in organic chemistry [29], for details see Appendix C. We compared the method's findings against the expert model. The expert rules were made following all the conventions and accommodating systematic chemical theory. The data-driven model had a limited data set with structures and labels only. Yet by far and large, the data-driven conclusions are in line with the expert rules. Sometimes, the expert model uses parameters, other than presence/absence of a substructure, for example, a solubility-related coefficient *log p*. These parameters, too, can be successfully predicted from structure, and the *Bridging Approach* copes in the absence of this knowledge. Further details on the model comparison are given in Appendix C.

5. Conclusions

Our idea has been to bridge from syntactic to statistical pattern recognition. The feature vector is filled with the counts of the substructures that are extracted from grammars. Having grammars automatically solves the task of feature segmentation and the choice of degree of abstraction for their representation. The selected features have sophisticated statistical properties, that is, max information gain at a particular point of the hierarchically divided feature space.

Compared to the syntactic paradigm, the new traits are:

- The proposed bridging model is directly recyclable in other applications, as long as the input can be parsed.
- It does not imply distance calculations and, instead, relies on vector classification, and, therefore, saves a non-trivial design step compared to the syntactic paradigm.

Having gained the new advantages, the method preserves the inherent strengths of the syntactic paradigm:

- Its essential capacity is to cope in the absence of expert knowledge, that is, no indications which features to extract.
- It preserves structural relations and works with a variable-length parsable input. It finds regions of interest in sequences with a suitable level of abstraction for their representation, and learns a decision tree that operates on presence/absence of these structures. Altogether, it leads to human-readable classification and gives insights into the problem.

6. Conflict of Interests

None declared.

7. Acknowledgements

As usually, we thank Torben Hagerup and Ricard Gavalda for thoughtful advice and guidance, – part of the work was completed while JS visited the LARCA at UPC. We acknowledge useful discussions with Antonio Berlanga, Florian Leitner, and Andrew Moss. JS acknowledges the *estancias postdoctorales* grant at the UC3M.

Appendix A. Parsing examples

Input to the parser is a SMILES of a chemical compound. The parser starts with the first atom in the SMILES string, uniquely identifies each atom with its position number and disambiguates which atom is linked to which other atoms and by which type of bond. Then, it reconstructs a tree representation of the compound.

An obvious challenge is that different SMILES exist for the same molecule. For example, a molecule from Figure 1 can be rotated and written as CCCCCSCC in place of CCSCCCCC. Canonical SMILES are not a solution, since they can't be drawn for the reason that are a hash value due to principles of their construction, and we don't want a black box construction. Our solution is sorting substructures in a natural order, when comparing sequences of substructures in grammar inference and in search.

Generally, the parser implements the SMILES language. Given the description from Section 2, few additional decisions are left to be made with

respect to how non-terminals are assembled and finally reduced to the start symbol S. The additional rewriting rules are as follows.

Rule 1: under our modeling non-terminals are denoted with the symbol sig_{arity} [5] and differ with respect to their $arity$, which is the number of the non-terminal's child nodes, for example:

$$sig_6 \rightarrow N1CCCCC1. \tag{A.1}$$

The numbers do not count, since they are special symbols.

Rule 2: Atomic chains, that is, molecules without rings or branches, are reduced directly to the start symbol S:

$$S \rightarrow CCSCCCCC. \tag{A.2}$$

Figure 1 depicts this molecule[6].

Rule 3: unlike atomic chains, branched and cycles are reduced to a non-terminal. For example, an atom and a branch hanging from it is reduced to a non-terminal of a corresponding arity. CC(CCCBr)CC is reduced to Csig_5CC, where $sig_5 \rightarrow$ C(CCCBr), as in Figure 5.

An example of multiple branches stemming from the same atom is CC(F)(Br)I, depicted in Figure 6.

Rule 4: children nodes of a non-terminal node can be atoms and/or substructures, and in order to calculate arity the number of such units is counted.

[5] Traditionally non-terminals are labeled with σ_{arity}, which we spell in the Latin alphabet as sig_{arity}.

[6] Also a functionality to depict SMILES can be useful, e.g. [35]

Rule 5: if a ring is not a stand-alone ring as in Figure 2, the starting atom of the cycle (the one after which the number is put) is marked as sig_1, and this disambiguates the branch from a cycle.

Very similar SMILES can lead to different parsing results, an example of the significance of parentheses is CC(C1CCC1)CC in Figure 7 (left) and CCC1CCC1CC in Figure 8.

Rule 6: In parser's output, shared atoms have a special tag that they belong to two different cycles. Otherwise, for example C2OC1CCC2CC1 in Fig 9, is simply reduced to $sig_6 sig_6$.

The above examples were simple to illustrate the parsing decisions. A couple of more complex molecules from the DSSTox NCTRER database are drawn in Figures 9 and 10.

Appendix B. Grammar Inference Algorithm

Under our modeling non-terminals are all marked with the symbol sig and differ with respect to their $arity$, which is the number of their child nodes, for example:

$$sig_6 \rightarrow N1CCCCC1. \tag{B.1}$$

The parsed SMILES are processed in postorder. In the algorithm below:

– j is the number of a node in post-order enumeration;

– X_j is a string of the child nodes of the node j:

$$X_j = x_1 x_2 ... x_l, \tag{B.2}$$

with $l \geq 1$. For example above, for sig_6 the string of child nodes is $NCCCCC$. Since the tree graph is traversed in post-order, at the point of reducing X_j to

a non-terminal sig_l, each of its child nodes $x_1 x_2 ... x_l$ have been parsed either as atoms or as non-terminal sig_{arity} nodes.

Data: The training set D of size n with parsed SMILES of chemicals belonging to the same activity class.

Initialization: Set G to contain empty sets for

- the set of atoms A,
- the set of non-terminals N,
- the set of rules P,
- the start symbol S,

and an empty table T with two columns: $\langle$ p: production from $P\rangle$ $\langle$ $count(p)$: how many times p was used$\rangle$.

$n =$ the number of instances in D.

for $i = 0$ **to** n, **while** $i < n$ **do**

$i = i + 1$;

read the i^{th} SMILES in D;

in postorder, **for** *each node* j *in* $SMILES$ **do**

If any atoms from X_j are not in A, add them to A.

If the string X_j can not be reduced with productions from $P\{$

add the rule: $sig_l \mapsto x_1x_2...x_l$ to P;

If sig_l is not in N, add it to N.

$\}$

Let p be the rule used to reduce X_j;

if p is not in T, add p to T with $count(p) = 0$.

$count(p) = count(p) + 1$;

end

end

Result: the grammar $G = (A, N, S, P)$ generalizing the activity class to which the input samples in D belong and a table T with grammar productions and their counts.

Algorithm 1: Polynomial time algorithm for grammatical inference of structures belonging to the same activity class from their parsed SMILES.

Appendix C. Data-driven and Expert Model

The data-driven model in the form of *if-, then-* rules obtained with the *Bridging Approach* is depicted in Figure 12. The conditions check for the presence of particular structural alerts. The ratio at the leaf boxes is ⟨number of correctly classified samples, number of errors⟩.

The expert model from [29] is summarized to:

1. If a chemical contains no ring structure, it is unlikely to be an estrogen receptor ligand (ER-ligand).
2. If a chemical has a nonaromatic ring structure, then it is unlikely to be an ER ligand, if it does not contain an O, S, N.
3. If a chemical has a non-OH aromatic structure, then its binding potential is dependent on the existence of key structural features and a solubility-related coefficient *log p*.
4. If a chemical contains a phenolic ring, then it tends to be an ER ligand, if it contains any additional key structural features. For the chemicals containing a phenolic ring separated from another benzene ring with the number of bridge atoms ranging from none to three, it will most likely be an ER ligand.

The rules in the expert model are based on systematic chemical knowledge. The data-driven model had a limited data set with chemical structures and labels for their activity only. Yet, by and large, the data-driven conclusions are in line with the expert rules:

1. Many of the chemicals covered by the 1st rule of the expert model end up at node 26 and node 31 passing as negative through numerous check-ups on the presence of different cyclic substructures.

2. The chemicals that satisfy the expert rule 7 from the original diagram [29] end up at the nodes 16 and 11. Node 9 (=C@, O) is equivalent to the presence of a phenolic ring.
3. The expert model has complex cases where the binding potential is determined with the help of non-structural information such as *log p*. The data-driven model is not allowed to use any additional information and accommodates these chemicals checking a lengthy list of structural conditions.

References

[1] M. Venguerov, P. Cunningham, *Generalized syntactic pattern recognition as a unifying approach in image analysis*, in: The 2nd international workshop on statistical techniques in pattern recognition, 1998, pp 913-920.

[2] R.G. Parekh, V. Honavar, *Grammar inference, automata induction, and language acquistion*, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, 2000, pp 727–764.

[3] C. De la Higuera, *A bibliographical study of grammatical inference*, Pattern Recognition 38 (2005) 1332-1348.

[4] E. Tanaka, *Theoretical aspects of syntactic pattern recognition*, Pattern Recognition 28 (1995) 1053-1061.

[5] M. Flasinski, J. Jurek, *Fundamental methodological issues of syntactic pattern recognition*, Pattern Anal. Appl. 17 (2014) 465-480.

[6] M. Ferrer, E. Valveny, F. Serratosa, H. Bunke, *Generalized median graph computation by means of graph embedding in vector spaces*, Pattern Recognition 43 (2010) 1642-1655.

[7] B. Luo, R.C. Wilson, E.R. Hancock, *Spectral embedding on graphs*, Pattern Recognition 36 (2003) 2213-2230.

[8] H. Bunke, K. Riesen, *Recent advances in graph-based pattern recognition with applications in document analysis*, Pattern Recognition 44 (2011) 1057-1067.

[9] B. Gauzere, P.A. Grenier, L. Brun, D. Villemin, *Treelet kernel incorporating cyclic, stereo and inter pattern information in chemoinformatics*, Pattern Recognition 48 (2015) 356-367.

[10] M. Vento, *A long trip in the charming world of graphs for Pattern Recognition*, Pattern Recognition 48 (2015) 291-301.

[11] J. Garcia, R. Aler, J. Sidorova, *Data Analysis*, Open Courseware Universidad Carlos III de Madrid (2014) https://www.youtube.com/watch?v=q4IHnJws3qE retrieved on 27/01/2015.

[12] J. Ernst, M. Kellis, *Discovery and characterization of chromatin states for systematic annotation of the human genome*, Nat. Biotechnol. 28 (2010) 817-825.

[13] A. Arcas, I. Cases, A.M. Rojas, *Serine/threonine kinases and E2-ubiquitin conjugating enzymes in Planctomycetes: unexpected findings*, Antonie Van Leeuwenhoek 104 (2013) 509-20.

[14] R. Durbin, S.R. Eddy, A. Krogh, G. Mitchison, *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*, Cambridge, UK, 1999.

[15] J.M. Sempere, D. Lopez, *Learning Decision Trees and Tree Automata for a Syntactic Pattern Recognition Task*, in: F.J. Perales et al. (Eds.), Pattern Recognition and Image Analysis, Springer-Verlag Berlin Heidelberg 2003, pp. 943-950.

[16] M.R. Ogiela, R. Tadeusiewicz, *Syntactic reasoning and pattern recognition for analysis of coronary artery images*, Artif. Intell. Med. 26 (2002) 145-159.

[17] J. Sidorova, A. Fernandez, J. Cester, R. Rallo, F. Giralt, *Predicting biodegradable quality of chemicals with the TGI+. 3 classifier*, in: The 11th IASTED International conference on artificial intelligence and applications, 2011, pp 108-115.

[18] J. Sidorova, M. Anisimova, *NLP-inspired structural pattern recognition in chemical application*, Pattern Recognit. Lett. 45 (2014) 11-16.

[19] I.H. Witten, E. Frank, M.A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edition, Morgan Kaufmann Publishers, 2011.

[20] T. Caetano, *The interplay of statistical and structural pattern recognition from a machine learning perspective*, in: International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2012, http://vimeo.com/38450616, retrieved on 27/01/2015.

[21] L. Rosenbaum, G. Hinselmann, A. Jahn, A. Zell, *Interpreting linear support vector machine with heat map molecule coloring*. Journal of Chemoinformatics 3 (2011): 11.

[22] D. Weininger, *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules*, J. Chem. Inf. Comput. Sci. 28 (1988) 31-36.

[23] R. Anand, K. Mehrotra, C.K. Mohan, S Ranka, *Efficient classification for multiclass problems using modular neural networks*, IEEE Transactions on Neural Networks 6 (1995) 117-124.

[24] T. Hastie, R. Tibshirani, *Classification by pairwise coupling*, Ann. Statist. 26 (1998), 451-471.

[25] T.G. Dietterich, G. Bakiri, *Solving multiclass learning problems via error-correcting output codes*, J. Artif. Intell. Res. 2 (1995) 263-286.

[26] M.A. Bagheri, Q. Gao, S. Escalera, *A genetic-based subspace analysis method for improving error-correcting output coding*, Pattern Recognition 46 (2013) 2830-2839.

[27] R. Kovahi, J.R. Quinlan, *Data mining tasks and methods: Classification:Decision tree discovery*, in: W. Klsgen, J.M. Zytkow (Eds.), Handbook of data-mining and knowledge discovery, Oxford University Press, Inc., New York, 2002, pp. 267-276.

[28] B. Hileman, *Hormone disrupter research expands*, Chem. Eng. News 75 (1997) 24-25.

[29] H. Fang, W. Tong, L.M. Shi, R. Blair, R. Perkins, W. Branham, B.S. Hass, Q. Xie, S.L. Dial, C.L. Moland, D.M. Sheehan, *Structure-activity relationships for a large diverse set of natural, synthetic, and environmental estrogens*, Chem. Res. Toxicol. 14 (2001) 280-294.

[30] R. Hannu, *In silico toxicology – non-testing methods*, Front. Pharmacol. 2 (2011) 33.

[31] N. Fonseca, V. Santos Costa, R. Camacho, *k-RNN: k-Relational nearest neighbor algorithm*, in: Symposium on applied computing (SAC), ACM, 2008, pp. 944-948.

[32] N. Landwehr, A. Passerini, L. De Raedt, P. Frasconi, *k-FOIL: learning relational kernels*, in: The 16th International Conference on Inductive Logic Programming (IPL), Short Papers (Muggleton,S. and Otero, R., eds.), 2006, pp. 125-127.

[33] A. Karwath, L. De Raedt, *SMIREP: predicting chemical activity from SMILES*, J. Chem. Inf. Model. 46 (2006) 2432-2444.

[34] H. Ahn, H. Moon, M.J. Fazzari, N. Lim, J.J. Chen, R. Kodell, *Classification from ensembles from random partitions of high-dimensional data*, Comput. Stat. Data An. 51 (2007) 6166-6179.

[35] N.M. OLBoyle, M. Banck, C.A. James, C. Morley, T. Vandermeersch, G.R. Hutchison, *Open Babel: An open chemical toolbox*, J Cheminf, 3 (2011) 33.

***Author Biography**

Julia Sidorova is a postdoctoral fellow at the Universidad Carlos III de Madrid, graduated in 2002 from Moscow State University and got her PhD from Universitat Pompeu Fabra in 2009. Research interests include pattern recognition, chemical activity prediction and affect recognition from voice.

Jesus Garcia is an associate professor at the Universidad Carlos III de Madrid, got his PhD from Universidad Complutense de Madrid. Research interests include Data and Information fusion, Multiagent systems and distributed sensor networks, Machine learning and data mining, Applied computational intelligence in engineering: Air Traffic Control, Surveillance and Machine Vision, Autonomous Vehicles Navigation & Control.

Table 1

Predicted as --	Predicted as inconclusive	Predicted as +	Real Class:
66	0	27	--
5	0	3	inconclusive
29	0	100	+

Table 1: Confusion matrix: active (+), inconclusive, and inactive (–).

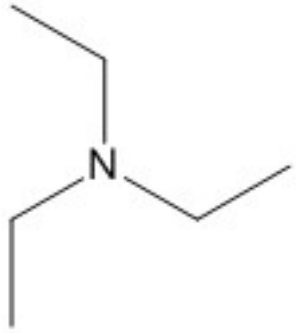
Figure 1

Atomic chain CCSCCCCC.

Figure 2

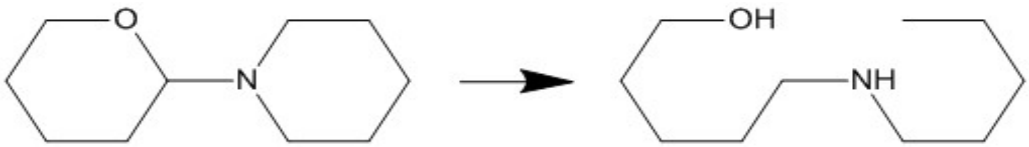
Double bond: C=C.

Figure 3



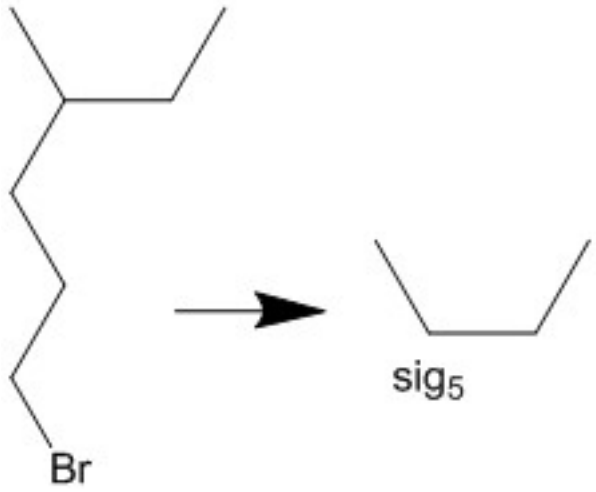
Branches: CCN(CC)CC.

Figure 4



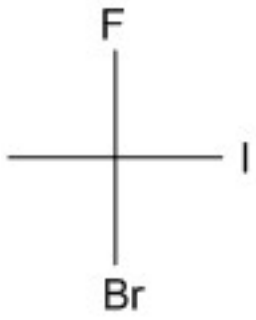
SMILES: O1CCCCC1N1CCCCC1. Rings are broken, and a number is put to leave a mark where the bond was broken.

Figure 5



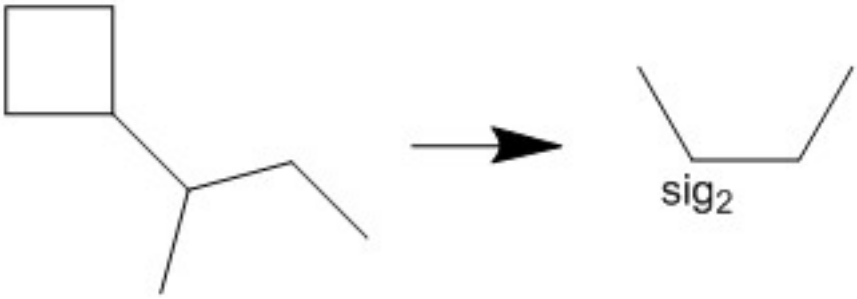
Branch is reduced to a non-terminal.

Figure 6



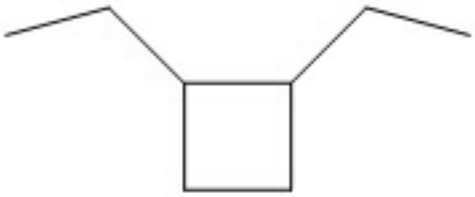
CC(F)(Br)I is reduced to “Csig$_3$I”.

Figure 7



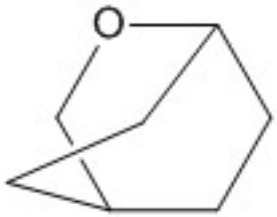
SMILES CC(C1CCC1)CC. The cycle is reduced to a non-terminal sig_2. The nonterminal has two child units: an atom and another non-terminal.

Figure 8



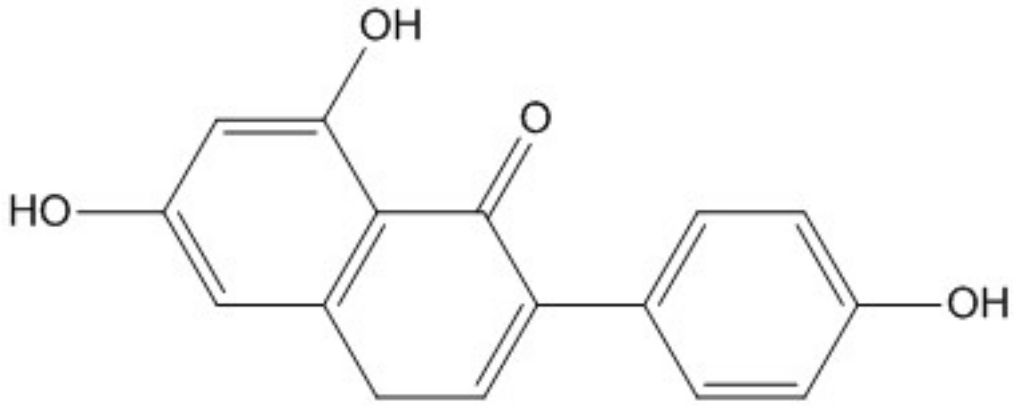
SMILES: CCC1CCC1CC.

Figure 9



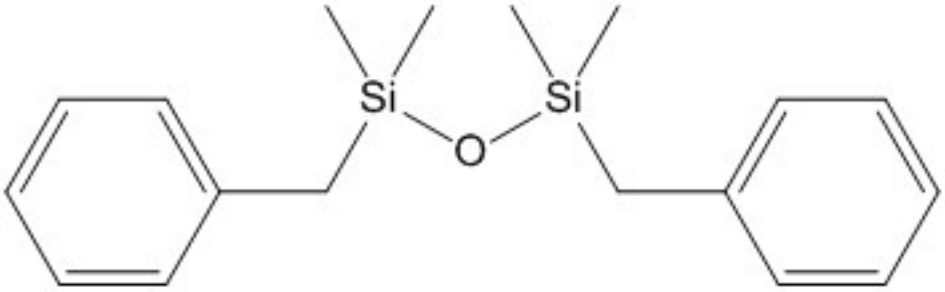
Two intersecting rings: C2OC1CCC2CC1. In the parser's internal presentation the atoms there are special tags @1 (and @2) after each atom, disambiguating to which cycle it belongs. Shared atoms are followed by @1@2:
C$_1$@1@2**O**$_2$@1@2**C**$_3$@1@2**C**$_4$@2**C**$_5$@2**C**$_6$ @1@2**C**$_7$@1**C**$_8$@1. Subscript is atom's ID from the SMILES strings.

Figure 10



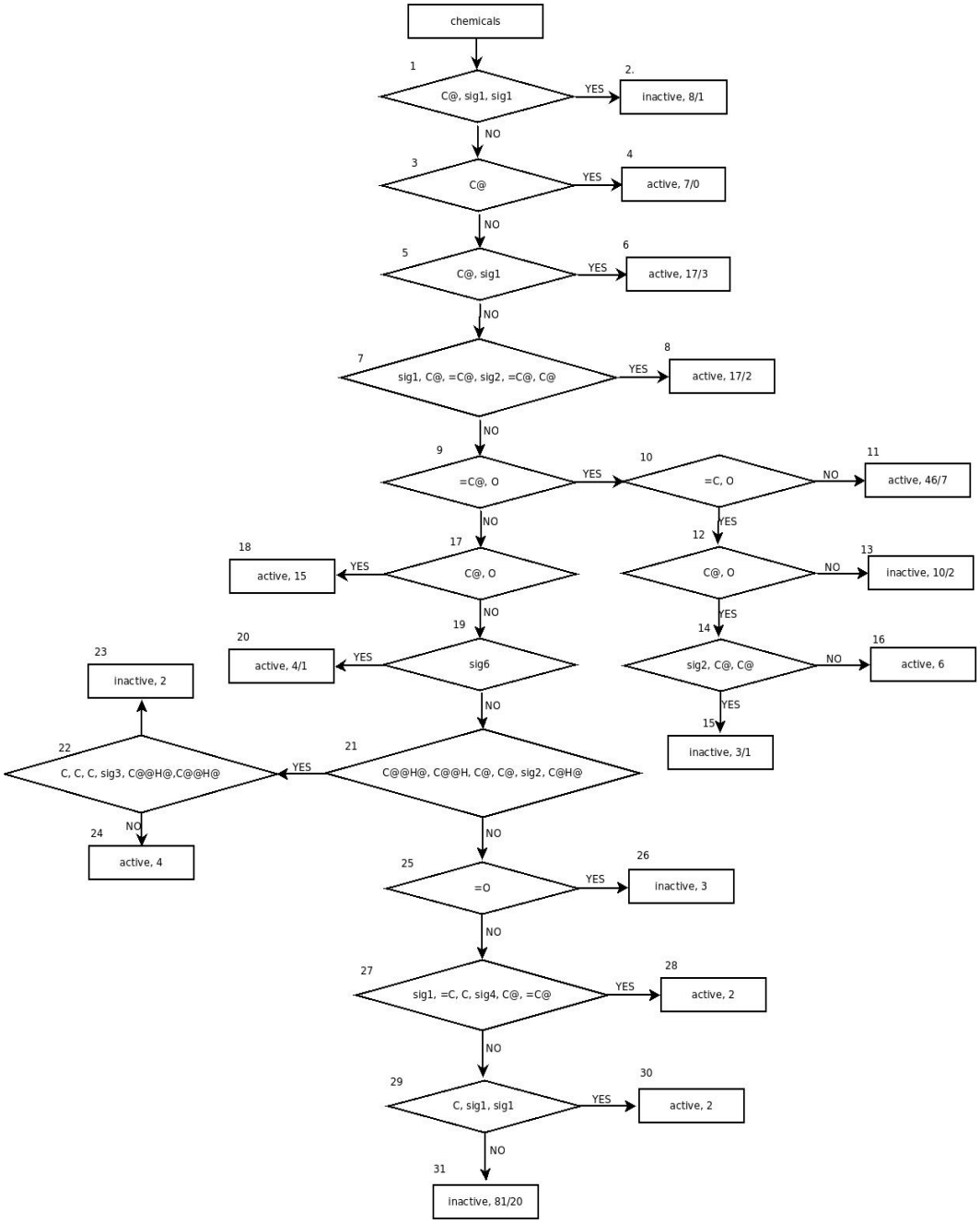
A molecule from the database:
O=C(C(C(C=C3)=CC=C3O)=CO2)C1=C2C=C(O)C=C1O.

Figure 11



A molecule from the database:
O([Si](CC1C=CC=CC=1)(CC)[Si](CC2=CC=CC=C2)(C)C.

Figure 12



Method's predictive model in the form of IF- THEN- rules.