REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From – To)* 1 July 2005 - 1 January 08

4. TITLE AND SUBTITLE Computer-Aided Synthesis Design of Energetic Compounds	**5a. CONTRACT NUMBER** FA8655-03-D-0001, Delivery Order 0022
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Vladimir A. Palyulin	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Moscow State University Leninskie Gory, 1/3 Moscow 119992 Russia	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD Unit 4515 BOX 14 APO AE 09421	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) EOARD Task 04-9010

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The aim of computer-aided synthesis planning is to assist organic chemist in planning synthetic work. Non-empirical approaches to computer synthesis use a small number of very general reaction principles for deriving chemical reactions, while empirical approaches rely on databases of known organic reactions. A major drawback of non-empirical approaches lies in impossibility to assess plausibility of derived organic reactions without an explicit interference of human expert (experienced organic chemist). A shortcoming of empirical approaches lies in their too strict reliance on sets of organic reactions kept in databases, which usually encompass only manually prepared well-known and well-characterized chemical transformations. The efforts in the framework of the present project are based primarily on the empirical approach to computer-aided organic synthesis planning. However, the underlying database of experimental reactions should contain information on a substantial number of reactions. The generalized reaction patterns, rules and reaction principles are automatically extracted by means of data mining informational techniques in order to support inferring such transitions of chemical compounds that are not explicitly contained in databases..

15. SUBJECT TERMS

EOARD, synthetic methods, computer synthesis, organic chemistry

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Brad Thompson
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	UL	38	**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)1895 616163

CRDF PROJECT# RUC1-1503-MO-05

Computer-Aided Synthesis Design of Energetic Compounds

Project Director: Vladimir Palyulin
Principal Organization: Department of Chemistry,
Moscow State University

Final Technical Report

Reporting Period: July 1, 2005 – November 30, 2008

Nomenclature

AAM – Atom-to-Atom Mapping
AUC – Area Under Curve
CCES – connected common edge subgraph
CR – concrete reaction
CS – computer synthesis
DB – database
ERC – extended reaction core
ERC_D (D=integer) – depth-extended reaction core
ERC_M – multi-step extended reaction core
ERC_O – one-step extended reaction core
ERC_S – smallest extended reaction core
ERC_T – two-step extended reaction core
FN – (number of) false negatives
FP – (number of) false positives
FP – structure fingerprint
FPR – false positive rate
HOC – Hierarchically Ordered Extended Connectivities
ID – identifier
MCES – Maximum Common Edge Subgraph
MCES – maximum common edge subgraph
MCIS – Maximum Common Induced Subgraph
MCS – Maximum Common Subgraph
MCS – maximum common subgraph
pseudoreaction – reasonable but failing (or low-yield) potential reaction
RC – reaction core
RDBMS – relational database management system
RDF – reaction data file
ROC – Receiver Operator Characteristic
RT – reaction type
SDF – structure data file
SRFP – structural reaction fingerprint
SSSR – Smallest Set of Smallest Rings
SVM – Support Vector Machines
TN – (number of) true negatives
TP – (number of) true positives
TR – generalized reaction transformation
transformation product – fragment of a product structure included in the transformation
transformation reactant – fragment of a reactant structure included in the transformation

The names and identifiers of the program objects (variables, functions, databases, tables, etc.) are marked in *italic* to differentiate them from the descriptive text.
Code fragments are shown with `Courier` font.

Summary

The aim of computer-aided synthesis planning [1] is to assist organic chemist in planning synthetic work. Non-empirical approaches to computer synthesis use a small number of very general reaction principles for deriving chemical reactions, while empirical approaches rely on databases of known organic reactions. A major drawback of non-empirical approaches lies in impossibility to assess plausibility of derived organic reactions without an explicit interference of human expert (experienced organic chemist). A shortcoming of empirical approaches lies in their too strict reliance on sets of organic reactions kept in databases, which usually encompass only manually prepared well-known and well-characterized chemical transformations. The efforts in the framework of the present project are based primarily on the empirical approach to computer-aided organic synthesis planning. However, the underlying database of experimental reactions should contain information on a substantial number of reactions. The generalized reaction patterns, rules and reaction principles are automatically extracted by means of data mining informational techniques in order to support inferring such transitions of chemical compounds that are not explicitly contained in databases.

In the course of a project, algorithms and software modules supporting all required functions of the synthesis planning software were developed. The MySQL database schema allows for efficient storage and fast retrieval of information concerning organic reactions, including chemical structures of reactants and reaction products, reaction conditions, yield, and literature references, as well as the relations between generalized reactions of different levels. A set of flexible algorithms and software modules for manipulating molecular and reaction graphs provides fast solution of graph-theoretical problems occurring in the synthesis planning.

A central role in the computer-aided synthetic and retrosynthetic analysis belongs to the generic reactions representing certain common reaction patterns. To take into account different hierarchical levels of specificity and generalization, the following types of generic reactions may be considered: Concrete Reaction (full chemical reaction with specific atom-to-atom mapping of reactants and products), Reaction Type (generic reaction pattern including all the reaction centers), Reaction Core (smallest connected graph including the RT pattern), Extended Reaction Cores (RC pattern augmented by certain neighboring atoms), Generalized reaction transformation (connected labeled graph derived from the set of the reaction center atoms). The procedures for automated detection of the reaction centers and recognition of the various levels of generic reactions from the raw chemical reactions reported in the synthetic literature were developed in the course of the project.

The database of organic reactions and transformations oriented to the synthesis of energetic compounds was created based on the literature data. It contains a significant number of reactions collected from literature, including a set of most important general-purpose organic reactions as well as specialized sets of reactions leading to specific classes of compounds. From 821 raw reactions contained in the database, a total number of 2596 generalized transformations were derived at different levels of generalization. For each reaction, the information on reaction conditions, literature references and yield (where available) is also stored. All the reactions can be easily browsed from within the synthesis planning software.

The algorithms and routines for synthetic and retrosynthetic analyses were developed. They support detection of possible reactions, construction of the respective reagent structures, checking of the valence constraints and generation of visual representation of the reaction. Each proposed precursor compound can, in turn, serve as a target compound for the next step of the retrosynthetic analysis. The testing sows that the software can be employed to suggest reasonable synthetic routes to compounds of interest.

In most non-trivial cases, many different precursors and synthetic paths are possible for a given target compound. Some of them are more realistic than the others, and the techniques for the quantitative assessment of plausibility of potential organic reactions and reaction paths proposed by the synthetic and retrosynthetic analysis routines are required. Substantial effort was devoted to this problem, and two approaches to such assessment were proposed and implemented in the synthesis planning software. The first approach involves the analysis of structural similarity of proposed reaction to known successful reactions using the Tanimoto similarity indexes of their structural reaction fingerprints. Another approach is based on the statistical modeling of the patterns common to the successful reactions in contrast to related reasonable but failing reactions (pseudoreactions).

To complement the empirical approach to computer synthesis, the non-empirical computer synthesis module was developed. It is intended primarily as a ‘fall-back’ for the cases where suitable synthetic approaches cannot be found by the empirical synthesis module using the available reaction data.

The user-friendly computer-aided synthesis planning software created in the course of a project provides all the features required for synthetic design. It is developed in the Java programming language and can be used on a wide variety of hardware and software platforms. Among its major features are: import and direct entry of experimental reaction data (including information on conditions, yields and literature references); automatic construction of the hierarchy of generalized transformations; browsing and editing of the reaction data; determination of the potential synthetic approaches and precursor structures by means of retrosynthetic analysis based on the available reaction database; browsing of information on similar experimental reactions (including conditions and literature references) for each proposed reaction; ranking of plausibility of the proposed reactions using reaction similarity index or statistical model; planning and plausibility assessment of the multi-step reaction paths; non-empirical synthesis planning based on a small number of reaction principles.

Thus, in the course of a project, a substantial progress in the field of computer-aided synthesis planning was made. Major theoretical problems were solved, and efficient algorithms and software modules supporting all steps of the planning process were developed and implemented in the user-friendly cross-platform computer-aided synthesis planning software. The testing shows that it can be used to suggest reasonable synthetic routes to compounds of interest.

Recommended areas of future development of this approach and the computer-aided synthesis planning system include: extension and refinement of the reaction and transformation database; development of advanced statistical techniques for the statistical evaluation of plausibility of the reactions and reaction paths; integration of the neural-network modules for the prediction of a number of relevant physico-chemical properties of target compounds; convenient manipulation and analysis of the retrosynthetic trees in order to identify the most promising approaches.

As a result of additional effort, the computer-aided synthesis planning software will allow the researcher to obtain more realistic synthesis proposals based on richer set of available techniques as well as quickly and easily estimate the relevant physico-chemical properties of target compounds.

Introduction

The aim of computer synthesis (CS) is to assist organic chemist in planning synthetic work. CS can operate in two directions: direct and reverse. Direct CS finds plausible synthetic routes starting from a given compound (or reaction products for given chemicals, in the simplest one-step case), while the reverse CS finds synthetic routes leading to given compounds. In other words, the direct CS answers the question as to what can be synthesized from a given compound, while the reverse CS answers the question as to how to synthesize it.

All CS approaches can roughly be subdivided into two main types: non-empirical and empirical. Non-empirical approaches use a small number of very general reaction principles expressed in the terms of the graph theory or Ugi-Dugundji BE-matrix formalism for deriving chemical reactions, while empirical approaches rely on databases of known organic reactions. The most known computer programs for empirical CS are LHASA, SECS, WODCA and CAMEO, while the list of non-empirical CS programs includes EROS, FLAMINGOES, COMPASS, etc. A major drawback of non-empirical approaches lies in impossibility to assess plausibility of derived organic reactions without an explicit interference of human expert (experienced organic chemist). As a result, non-empirical CS approaches can advantageously be used only for designing general synthetic approaches. A shortcoming of empirical approaches lies in their too strict reliance on sets of organic reactions kept in databases, which usually encompass only manually prepared well-known and well-characterized chemical transformations.

The efforts in the framework of the present project are based primarily on the empirical approach to CS. To overcome its drawbacks, two novel features were proposed. Firstly, the underlying reaction databases should contain information on a substantial number of reactions reported in the literature. This would allow introducing non-trivial transformations into CS. The second important feature is the application of data mining informational techniques for automatic extraction of generalized reaction patterns, rules and reaction principles in order to support inferring such transitions of chemical compounds that are not explicitly contained in databases.

Technical Description of Work

In accordance with the originally established plan, the work in the course of a project pursued the following general directions:

- Development of the database schema for efficient storage and manipulation of reaction and transformation data.
- Development of algorithms and software modules for the efficient manipulation of chemical structures and reactions.
- Development of approaches and software modules for automated extraction of generalized organic transformations from reaction data.
- Creation of the database of organic reactions and transformations oriented to the synthesis of energetic compounds.
- Development of algorithms and software modules for synthetic and retrosynthetic analysis.
- Development of approaches to assessing the plausibility of proposed reactions and reaction paths.
- Development of algorithms and software modules supporting non-empirical approach to retrosynthetic analysis.
- Development of the integrated, user-friendly synthesis planning system that implements all the required functionality.

Let us consider in detail the major results achieved in each of these areas.

Development of the database schema for efficient storage and manipulation of reaction and transformation data

The database schema and software modules for storage and manipulation of the organic reaction and transformation data allow for efficient storage and fast retrieval of information concerning organic reactions, including chemical structures of reactants and reaction products, reaction conditions (temperature, solvents, additional reagents and/or catalysts, *etc*), yield, and literature references. In addition, the relations between the raw and generalized reactions of different levels are also handled. If necessary, the database schema can be expanded to support additional attributes of chemicals and reactions.

The reaction database manipulation system is based upon the MySQL open-source relational database management system (RDBMS). The database encompasses 7 tables that contain different levels and types of information: *atom*, *bond*, *comment*, *comment_type*, *reaction*, *reaction_relation*, *structure*. Let us describe each table schema in more detail. (Several other tables containing some auxiliary data required for the operation of the synthesis planning software are also present.)

The *reaction* table is used to store the identifiers of reactions present in a database. It contains the following fields (SQL field type is listed in parentheses):
id (int) – unique ID of a reaction record;
reaction_type (int) – type of reaction (10 – reaction as transformation, 200-299 – generic reactions at different levels, 300 normal reaction);
ctime (timestamp) – creation date/time for a reaction record;
mtime (timestamp) – modification date/time for a reaction record.

The *structure* table is used to store the identifiers of reactant and product structures as well as auxiliary data used in fast substructure search procedure. It contains the following fields:
id (int) – unique ID of a structure record;
reaction_id (int) – ID of a reaction record (*id* field in *reaction* table) involving a given structure;

structure_type (int) – structure type (role in reaction): 0 – reactant, 1 – product, 10 – reaction as transformation;
structure_dump (blob) – structure dump: serialized representation of the molecular graph of a structure containing only atom type and bonding data (see below);
fingerprint1-fingerprint8 (bigint) – eight 64-bit fields used to store structure fingerprint (see below).

The *atom* table is used to store the information on the specific atoms of each structure. It contains the following fields:
id (int) – unique ID of an atom record;
reaction_id (int) – ID of a reaction record (*id* field in *reaction* table) involving a given atom;
structure_id (int) – ID of a structure record (*id* field in *structure* table) involving a given atom;
atom_num (int) – sequential number of an atom within a structure (1-based);
x (double) – atom X coordinate;
y (double) – atom Y coordinate;
z (double) – atom Z coordinate;
symbol (varchar(3)) – chemical symbol of an atom;
charge (int) – formal charge of an atom;
isotope (int) – isotope mass number or 0 if no isotope is explicitly specified;
symbol_int (int) – atomic symbol packed into integer;
symbol_charge_int (int) – atomic symbol and charge packed into integer;
radical (int) – radical center;
stereo_parity (int) – stereo center parity: 0 – not specified, 1 – odd, 2 – even, 3 – either;
hydrogen_count (int) – number of attached hydrogen atoms;
stereo_care_box (int) – flag for matching double bond stereo configuration;
valence (int) – atom valence: number of attached bonds (including implicit hydrogens).

The *bond* table is used to store the information on the specific bonds between atoms of each structure. It contains the following fields:
id (int) – unique ID of a bond record;
reaction_id (int) – ID of a reaction record (*id* field in *reaction* table) involving a given bond;
structure_id (int) – ID of a structure record (*id* field in *structure* table) involving a given bond;
atom_1_num (int) – sequential number of the first bonded atom within a structure (1-based);
atom_2_num (int) – sequential number of the second bonded atom within a structure (1-based);
bond_type (int) – bond type: 1 – single, 2 – double, 3 – triple;
stereo (int) – bond stereochemistry: 0 – not specified, 1 – up, 4 – either, 6 – down;
bond_topology (int) – bond topology: 1 – ring bond, 2 – chain bond, 0 – either (reserved for future use, default = 0);
reacting_center_status (int) – status of a reacting center.
markush (int) – variant of a Markush formula [2] (0 for normal non-Markush bond).

The generalized transformations are stored using the same database schema (in particular, the tables *reaction*, *structure*, *atom*, *bond*). Generalized transformation graph is stored as a structural formula of reagent (in table *structure*, the *structure_type* field equals 0). The level of a generic reaction (see below) is encoded by the *reaction_type* field in *reaction* table. The relations between the reactions of different levels (specific reaction -> generic reaction) are stored as records in the *reaction_relation* table. It contains the following fields:
id (int) – unique ID of a relation record;
reaction_1_id (int) – ID of record for the 1st reaction (*id* field in *reaction* table);
reaction_2_id (int) – ID of record for the 2nd reaction (*id* field in *reaction* table);
relation_type (int) – type of relation between reactions (reserved).

The *comment* table is used to store comments – different types of textual data related to a reaction (e.g., reaction conditions, literature references, etc.). It contains the following fields:
id (int) – unique ID of a comment record;
reaction_id (int) – ID of a reaction record (*id* field in *reaction* table) related to a given comment;
comment_type_id (int) – type of comment data (*id* field in *comment_type* table);
comment (text) – the contents of a comment.

The *comment_type* table is used to store definitions of different comment types. It contains the following fields:
id (int) – unique ID of a comment type record;
type_name (varchar(255)) – verbal description of a type;
type_code (varchar(255)) – type code (e.g., LITTEXT).

As explained above, at the database level individual atom and bond records are entirely independent and can be easily retrieved by a query matching reaction and structure ID fields. Thus, DB row size is independent of the structure size. Moreover, the complete atom and bond information can be obtained without unpacking some encoded structure representation.

It is complemented by the *structure dump* – a compact serialized representation of a molecular graph based on its connectivity. Dump is an array of atom objects, each containing 5 fields (Java variable type is listed in parentheses):
symbol (int) – atomic symbol packed into integer;
originalNum (int) – original position of a given atom within a structure;
degree (byte) – number of attached bonds for a given atom;
bondTypes (byte[]) – types of bonds for a given atom;
atomNumbers (int[]) – sequential numbers of atoms (array indices) for each bonded atom.

The dump is constructed in such a way that each dump atom (except the first one) is connected to at least one atom preceding it in the array. (Consequently, atom order in dump may differ from the original structure). This property enables the substantial reduction of enumerative search during the detection of a given fragment within a molecular graph.

Structure fingerprint (FP) [3] is a 512-bit binary string. When constructing a fingerprint for a given structure, a number of substructures are identified including atoms, bonds, paths of length 3..7, cycles up to 6 atoms, and star-type fragments containing central atom and 3-4 neighboring atoms [4]. Each substructure is used to initialize a random number generator yielding substructure fingerprint – 512-bit binary string containing 3 unity bits. Substructure fingerprints are merged by means of a bitwise OR operation yielding full structure fingerprint. For each structure (FP1) and substructure (FP2) fingerprints the following property holds: if FP2 contains 1 in position *i* then FP1 also will contain 1 in position *i*. This enables the fast pre-screening of the large structure and reaction series for the presence of given substructures.

The synthesis planning system developed in the course of a project supports populating the reaction database (based on the reactions entered directly or imported from an external RDF file [5]), including detection of generalized transformations. In addition, it is possible to browse the reaction database as well as to perform a *substructure search* in order to select for browsing only reactions where reactants or products contain a specified molecular fragment.

Development of algorithms and software modules for the efficient manipulation of chemical structures and reactions

Flexible and efficient tools for manipulating molecular and reaction graphs play a fundamental role in any computer synthesis system. Algorithms solving some problems are quite straightforward. On the other hand, certain problems belong to the NP-complete class and require a number of preliminary tests in order to reduce the search space to tractable size.

In particular, the following algorithms and implementations are required:

- Construction of the adjacency matrix of a molecular graph (encoding atom and bond types) from the lists of molecule atoms and bonds
- Detection of aromatic bonds in the ring systems (from the common representation as a set of alternating single and double bonds)
- Check of connectivity of a molecular graph and detection of connected components
- Determination of the topological equivalence classes for vertices of a molecular graph
- Detection (check) of graph isomorphism
- Substructure search (subgraph isomorphism)
- Detection of the maximum common subgraph (isomorphic intersection)
- Detection of rings (cycles) in a molecular graph
- Determination of the automorphism group of a molecular graph

To achieve these goals, we developed a number of routines for manipulating molecular and reaction graphs based either on the algorithms developed by us or on the modified versions of the algorithms published in the literature. In most cases, hydrogen-suppressed molecular graphs are considered (unless stated otherwise). In complex tasks (e.g., isomorphism/automorphism analyses) several steps of preliminary filtering can be employed to enhance the performance, based upon structure fingerprints, size of the graphs, topological equivalence classes, etc.

Construction of the adjacency matrix of a molecular graph

This procedure builds the adjacency matrix from the lists of molecule atoms and bonds (hydrogen atoms are ignored).

Non-diagonal elements of the matrix $A_{ij}=A_{ji}$ define the type of a bond between the *i*-th and *j*-th vertices of a molecular graph:

0 – no bond; 1 – single bond; 2 – double bond; 3 – triple bond; 4 – aromatic bond

Diagonal elements A_{ii} define the type of an atom corresponding to the *i*-th vertex of a molecular graph.

Detection of aromatic bonds

In reaction databases, the aromatic bonds are usually not marked explicitly. Instead, they are represented as a set of alternating single and double bonds. Thus, a detection procedure for the aromatic bonds is required as their presence may significantly affect the synthetic and retro-synthetic analysis. The routine employed is based on the backtracking algorithm. By scanning the cycles sequentially, it analyzes the bonds in them and marks the bonds found to be aromatic. The analysis of a molecular graph is repeated iteratively until no new aromatic bonds are detected during a given iteration. The routine enables the detection of aromatic bonds in monocyclic as well as polycyclic systems.

Check of connectivity of a molecular graph and detection of connected components

The procedure performs a recursive walk of a molecular graph in the order of vertex adjacency (starting with the first vertex) and labels each vertex as belonging to the first connected component. Then the list of graph vertices is sequentially scanned for unlabelled vertices. If such an unlabelled vertex is found, the similar routine is repeated starting from it (and the vertices are labeled as belonging to the second connected component), and so on.

Determination of the topological equivalence classes for vertices

The procedure performs a complete discrimination of atoms into classes of topological equivalence (graph orbits). This information is used later in order to substantially enhance the performance of graph isomorphism analysis.

The routine is based on the HOC (Hierarchically Ordered Extended Connectivities) algorithm [6]. At the end, each vertex of a molecular graph is assigned a class label (from 1 to *k*, where *k* is a number of classes of topological equivalence found in a structure).

Detection of the graph isomorphism

The graph isomorphism routine checks whether the two given graphs are the same (with possibly different numbering of vertices).

The procedure employs a number of successively more detailed stages of filtering in order to avoid very expensive direct vertex-by-vertex checking of the isomorphism as much as possible [7-12].

1) Graphs having different number of vertices and/or edges cannot be isomorphic, and the procedure returns negative result.

2) Determine the topological equivalence classes for both graphs. In isomorphic graphs the vertices that can be mapped to each other should belong to the same topological equivalence class. If the number of classes for the two graphs is different, these graphs cannot be isomorphic, and the procedure returns negative result.

3) For each pair of the same topological equivalence classes, check whether the class vertices have the same label (atom type); whether these classes contain the same number of vertices; whether their closest environment is the same (taking into account the topological equivalence classes and bond types for the directly connected vertices). If a mismatch is found at any step, these graphs cannot be isomorphic, and the procedure returns negative result.

4) Perform the detailed check of graph isomorphism. Using the backtracking algorithm, an attempt is made to find a vertex-to-vertex mapping between the two graphs. To reduce the search space, the data on the topological equivalence classes of the vertices is used. If such a mapping is found, the graphs are isomorphic.

Substructure search

In graph-theoretical terms, the substructure search is equivalent to the problem of Subgraph Isomorphism [13-15] that involves the checking whether one graph is completely contained within another, larger one (such that an isomorphic mapping exists between the search structure and a certain subgraph of a larger graph).

We have developed and implemented the algorithm for substructure search that combines backtracking and molecular fingerprints [3]. During the search, the dump and fingerprint are first constructed for each such fragment. Preliminary screening (filtering) of reactions is performed using the following SQL query:

```
SELECT id, reaction_id, structure_dump FROM structure WHERE fingerprint1&{fp1}={fp1} AND
fingerprint2&{fp2}={fp2} AND ... fingerprint8&{fp8}={fp8}
```

where *`{fp1}`*, *`{fp2}`* ... *`{fp8}`* – values of the 1st, 2nd, ..., 8th 64-bit part of the fragment fingerprint.

If the user is interested in a fragment search for reactants only, the SQL query is augmented by the string `" AND structure_type=0"`. Similarly, for product-only search, the string `" AND structure_type=1"` is added.

For resulting records, the *structure_dump* field value is de-serialized and matched against the fragment dump using a backtracking procedure. If a fragment is detected as a subgraph of a structure, the *reaction_id* identifier is added to the list of substructure search results.

Detection of the maximum common subgraph (MCS)

The goal of maximum common subgraph search is to identify in two graphs the largest subgraphs that can be mapped isomorphically to each other [16]. Two types of MCS problem can be considered depending on the size measure used [17]. Maximum common induced subgraph (MCIS) has the maximum number of vertices while maximum common edge subgraph (MCES) has the maximum number of edges. In chemical applications, MCES is usually better suited to the study of molecular graph similarity [18]. MCES is a subgraph having the maximum number of edges common to the both graphs. In its turn, two types of MCESs can be considered: connected MCES and disconnected MCES. In connected MCES there is at least one path connecting any pair of vertices. Disconnected MCES can consist of two or more connected components.

Strictly speaking, MCS search is a so-called NP-complete problem – that is, a sequential algorithm for solving it in polynomial time is not currently known and likely does not exist at all. Available algorithms [19-21] can be classified into exact (using various techniques to reduce the search space) and approximate (using genetic algorithms and similar approaches to obtain close approximation to an answer in reasonable time by some sort of sampling of the search space). We have implemented the exact MCES search procedure based on the backtracking algorithm presented in [19]. Depending on the options selected, it can provide either connected or disconnected MCES. In order to reduce the search space, the analysis of the partial topological equivalence classes is used.

Detection of rings (cycles) in a molecular graph

In most cases it is sufficient to find the so-called Smallest Set of Smallest Rings (SSSR) [22-26] as other possible cyclic substructures can be derived from it. Many different algorithms for the detection of SSSR were proposed in the literature. We have implemented the algorithm described in [22]. In contrast to many other approaches, it provides complete, exact and efficient solution.

Determination of the automorphism group of a molecular graph

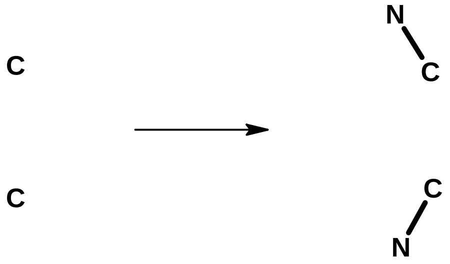
Graph automorphism is a permutation of the vertices that preserves their adjacency [27]. In other words, it maps edges of a graph to edges of the same type while non-edges (pairs of vertices without connecting edge) are mapped to non-edges. Our routine finds an automorphism group of a molecular graph as a set of possible permutations of all its vertices. For graphs having large number of vertices and high degree of symmetry the representation of an automorphism group as a list of its elements is problematic since group order may be as high as N! (where N is a number of vertices). Thus, in general case, it is better to determine the generating set of a group that always has no more than $N(N\text{-}1)/2$ elements [28]. However, in our work the primary application of the routine would be to determine the restriction of an automorphism group on a specific subset of vertices of a molecular graph (i.e., only permutations of these selected vertices are considered) since further analysis is concerned only with some specific substitution positions. As only a small number of substitution positions is to be considered in most cases (approximately 2 to 6), representation of an automorphism group as a list of permutations is preferred. At the preparation step of the routine, the set of the molecular graph vertices is split into the topological equivalence classes in order to reduce the search space and accelerate the analysis. Then the routine builds all the possible permutations (using the backtracking algorithms) and, for each permutation, tests whether the adjacency relation in the molecular graph is conserved.

Development of approaches and software modules for automated extraction of generalized organic transformations from reaction data

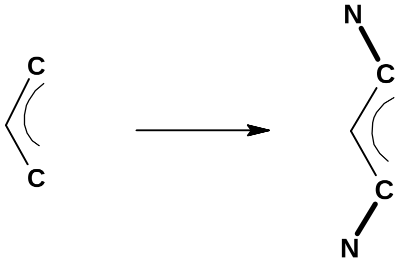
In the framework of the computer-aided synthetic and retrosynthetic analysis, central role belongs to the concept of generic reactions that represent certain common reaction patterns [1, 29-45]. Usually it is necessary to take into account several levels of specificity and generalization. The following levels may be considered:

Reaction centers – all atoms having at least one bond broken, formed or changed during the reaction.

Reaction type (RT) – generic reaction pattern including all the reaction centers. For the dinitration of *m*-diethoxybenzene, reaction type is as follows:



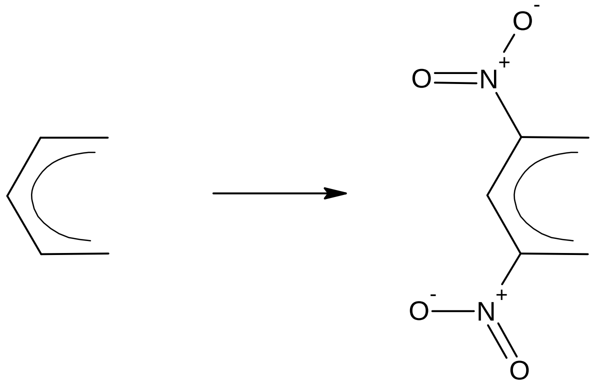
Reaction core (RC) – smallest connected graph including the RT pattern.



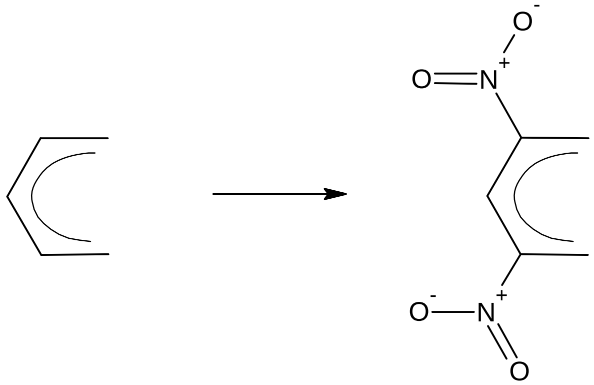
(Reaction centers and changed bonds are highlighted)

Extended reaction core (ERC) – RC pattern augmented by certain neighboring atoms. Several kinds of ERCs are used:

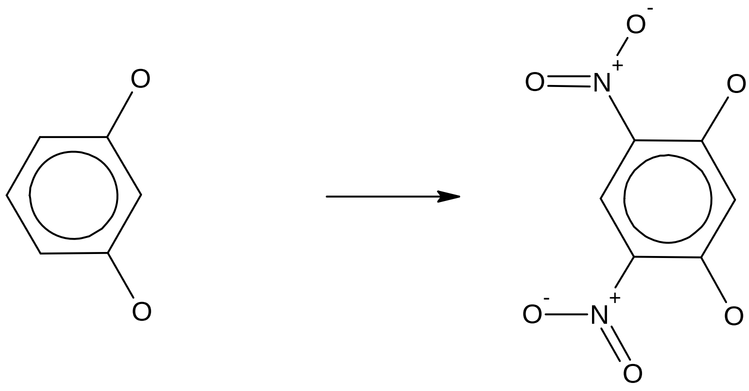
Smallest ERC (ERC_S) – RC augmented by the directly connected active atoms (that is, heteroatoms and carbon atoms connected to the RC by multiple bonds).



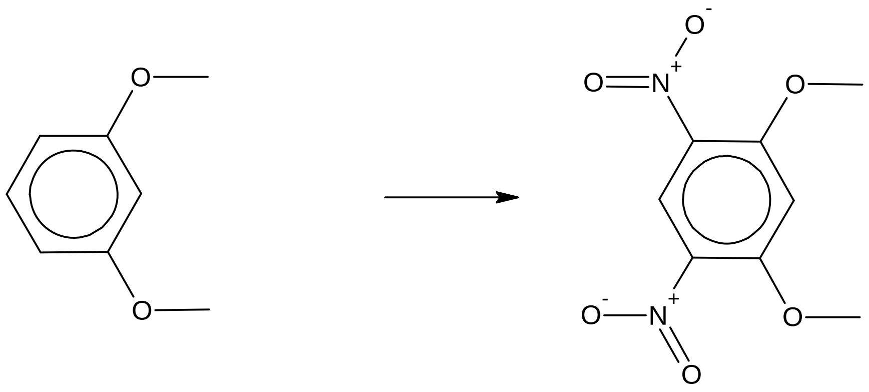
One-step ERC (ERC_O) – RC augmented by the directly connected non-aliphatic atoms (here, aliphatic atoms are carbon atoms connected only by single bonds to other carbon atoms).



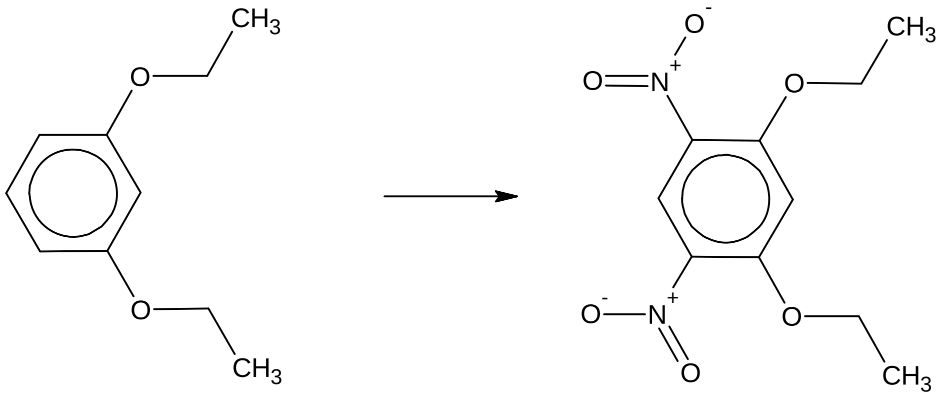
Two-step ERC (ERC_T) – RC augmented by the non-aliphatic atoms up to the distance of two bonds (here, aliphatic atoms are carbon atoms connected only by single bonds to other carbon atoms).



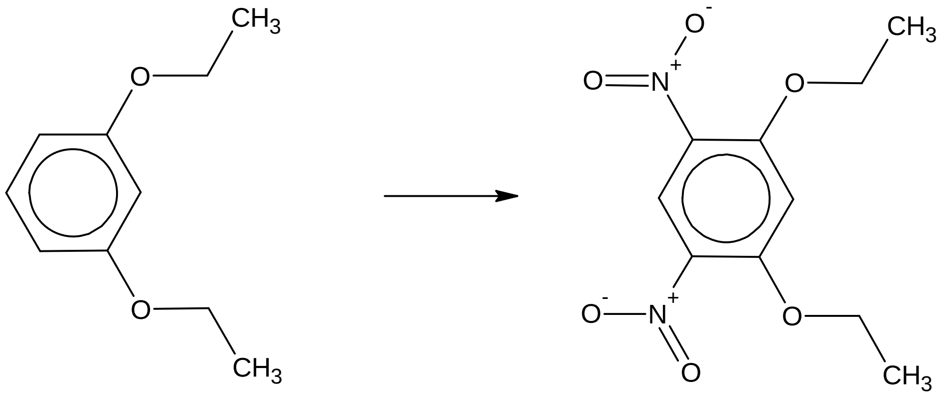
Multi-step ERC (ERC_M) – RC augmented by the non-aliphatic atoms up to the first aliphatic atom (here, aliphatic atoms are carbon atoms connected only by single bonds to other carbon atoms).



Depth-extended ERC (ERC_D, D=integer) – RC augmented by all non-hydrogen environment atoms up to the specified depth of D bonds. Below, the ERC_4 pattern for this reaction is illustrated.



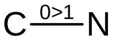
Concrete reaction (CR) – full chemical reaction with specific atom-to-atom mapping of reactants and products.



Some problems require the introduction of an additional reaction level – symbolic equation or generalized reaction transformation. In a sense, it is the most compact representation of a general reaction principle. Generalized reaction transformation (TR) is defined as a connected labeled graph derived from the set of the reaction center atoms. In contrast to the levels discussed earlier, the vertices in a generalized transformation graph superimpose the corresponding reagent and product atoms (only atoms and bonds belonging to the reaction center are considered). In addition, the vertex labels encode only broad classification of atoms rather than their exact chemical nature. Five classes are currently defined:

- C – carbon
- H – hydrogen
- Ha – halogen
- Me – metal
- Ot – all other elements (e.g., O, N, P and other heteroatoms)

The edge labels in a generalized transformation graph represent the changes of chemical bonds between the corresponding atoms in the course of reaction. They have the form “`x>y`” where `x` is the original bond order in the reagents and `y` is the final bond order in the products (0 – no bond, 1 – single, 2 – double, 3 – triple, a – aromatic). For instance, “`0>2`” is a formation of double bond and “`1>a`” is a transformation of single to aromatic bond. Thus, a generalized transformation graph captures the most important features of the chemical reaction in the synthesis planning context. For instance, for the *m*-diethoxybenzene dinitration process the reaction type (RT) contains all the reaction centers and no other atoms (see above). This transformation consists of three separate copies of the same elementary generalized transformation that occur in different parts of the molecule. The generalized transformation (TR) graph is the following:



In the context of synthesis planning, such a multi-level classification allows one to utilize the most detailed chemical knowledge available as well as to attempt transferring the reaction principles to other types of compounds.

The procedures for automated detection of the reaction centers and recognition of the various levels of generic reactions from the raw chemical reactions reported in the synthetic literature were developed in the course of the project. Several algorithms for the detection of reaction centers are available in the literature. However, analysis shows that they often fail to provide the correct solution of this problem, especially if (as is often the case for the organic reaction databases) the original raw reaction does not obey the stoichiometric balance (e.g., the molar ratio of reactants is specified incorrectly and/or some reagents are mentioned in the description of conditions rather than included in the equation). Thus, we had to develop a new method for the detection of reaction centers using some of the concepts proposed in the earlier literature.

Detection of the reaction centers

Starting from the raw chemical reaction, the detection of the reaction centers is performed using the following algorithm.

1) If several products are formed in the reaction, it is split into a set of simple reactions, each having exactly one product and the same reagents as the original reaction. In further analysis, only single-product reactions are considered.

2) Aromatic bonds are detected in the reagents and the product by means of the procedure described above.

3) Each reagent and product atom is assigned a unique number (atom label) that is used later to specify the atom-to-atom mapping of reagents to product.

4) A supplementary boolean array *product_atom_ban_list* is created with size equal to the number of product atoms. During the analysis, its elements show whether given product atom is already mapped to the corresponding atom in any of the reagents.

5) Atom-to-atom correspondence between reagents and product is determined using the procedure for the Maximum Common Subgraph (MCS) search. The MCS search for the molecular graphs takes into account the vertex labels (atom types). On the other hand, edge labels (bond types) are not considered because they can change in the course of reaction.

In the form of pseudo-code, the correspondence search procedure can be expressed as follows:

```
while(true)
{
  for(i=0; i<n; i++) // n – number of reagents
  {
    if(mapping is already found for the i-th reagent)
      continue; // proceed to the next reagent
    find the set of MCSs for the molecular graphs of the i-th reagent and the product (vertices marked in product_atom_ban_list are not
            considered);
    for(each MCS found)
      calculate match rating (+1 point for each match of vertex degrees, atomic charges and bond types in the molecular graphs);
  }

  if(no MCS found)
    break; // exit the while(true) loop

  choose MCS having the maximum rating;
  save the atom-to-atom mapping determined by the given MCS (assign respective reagent atom labels to the product atoms belonging to MCS,
        mark these atoms in product_atom_ban_list array to exclude them from the following iterations of the MCS search);
}
```

6) All bonds in the reagents and the product are analyzed. Using the atom-to-atom correspondence determined above, mark the bonds that are broken, formed or changed in the course of reaction.

7) Reaction data (including atom-to-atom correspondence and bond labels) is stored in the database as a CR-level reaction (type 200). Record specifying the relationship between raw and CR reactions is added to the *reaction_relation* table. Number of CR reactions corresponding to a given original reaction is equal to the number of its products.

Recognition of generic reactions

Various levels of generic reactions are determined from the CR reaction data.

To obtain the Reaction Type (RT) level, only atoms participating in the broken, formed and/or changed bonds are retained. Atoms having all other atom-to-atom mapping labels are removed.

Generic reactions of the Reaction Core and Extended Reaction Core levels (RC, ERC_S, ERC_T, ERC_M) are determined by the *getReactionCore* procedure having the variable parameter *level*. For RC *level = 0* , for ERC_S *level = 1*, for ERC_T *level = 2*, for ERC_M any sufficiently high number can be used. This procedure involves the following steps:

1) Reaction center vertices in the reagents and the product are marked as having zero distance from the reaction core.
2) Find the shortest paths connecting the above vertices in the reagent and product molecular graphs. Vertices belonging to these paths are also marked as having zero distance from the reaction core.
3) For all other atoms, the distance from the reaction core is calculated.
4) Vertices having distance from the reaction core greater than *level* value are removed.

The generic reactions obtained in this way are stored in the database. For each generic reaction, the references to the original reaction, CR reaction and higher-level generic reactions (e.g., ERC_T->ERC_M, ERC_S->ERC_T, ERC_S->ERC_M, etc.) are also added to the *reaction_relation* table. If an equivalent generic reaction is already present in the database, only the appropriate references are added.

Detection of generalized transformations
The procedure for the detection of generalized transformations uses as starting data the concrete reaction (CR) level representation – that is, the complete original reaction with atom-to-atom correspondence (*reaction_type* = 200).

The procedure is based on the following algorithm:
1. Aromatic bonds are detected in the reagent and product structures by means of the procedure described above.
2. Reaction center atoms are detected in the reagents and products using the procedure described above. By definition, atom belongs to the reaction center if it has at least one bond that is formed, broken or changed in the course of reaction. As a result, the list of the reaction center atoms labels (based on the atom-to-atom correspondence numbering) is obtained.
3. For each reaction center atom, the atom class is determined according to the classification defined above (C/H/Ha/Me/Ot). Then the vertex with a respective label is added to the transformation graph under construction.
4. For each vertex pair in a transformation graph, the algorithm determines the presence and type of bond between the corresponding atoms in the reagent and/or product structures. If the bonds in reagents and products are different, these vertices are connected by the edge with a label encoding the specific bond type change.
5. If the resulting raw transformation graph is not connected, it is split into the connected components and each component is analyzed as a separate generalized transformation.

This procedure defines only the topology of the transformation graphs (that is, the types and connections of vertices and edges). Since such a graph combines the atoms of different structures, meaningful transfer of atomic coordinates is not possible. In order to enable graphical display of the transformations, additional graph visualization step is required. The optimal set of vertex coordinates is approximated by means of a simplistic iterative algorithm. It models the dynamic behavior of a planar system of mutually repelling balls (graph vertices) that are connected by the rigid rods (edges). Initial coordinates are determined by the random number generator.

Thus, we have developed the software module allowing one to process the raw reaction data in the following way:
1) Split the original raw reaction into the single-product reactions.
2) Detect the reaction centers and store this information in the database as the CR level reaction.
3) For all CR reactions, determine and store the generic reactions at the ERC_M, ERC_T, ERC_S, RC, RT and TR levels, as well as the reaction relation references.

As a result, for a body of original raw reactions, a hierarchy of generic reactions is built that is used in synthetic and retrosynthetic analysis.

Creation of the database of organic reactions and transformations oriented to the synthesis of energetic compounds

For the development and testing purposes, a database of reactions suitable for the synthesis of nitro and azido compounds was constructed based on the literature analysis. In the final version of the software, the database contains a significant number of reactions collected from different sources. Currently the following reaction subsets are included:

- Small set of most important general-purpose organic reactions [46] (255 reactions)
- Selected set of representative reactions leading to nitro compounds [47] (35 reactions)
- Selected set of representative reactions leading to azido compounds (34 reactions)
- Reactions leading to furazan and furoxan compounds [48] (322 reactions)
- Reactions leading to triazole and tetrazole compounds [49, 50] (193 reactions)

Total number of raw reactions in the database is 821, number of concrete reactions (one per product) is 840. From them, a total number of 2596 generalized transformations were derived at different levels of generalization.

For each reaction, the information on reaction conditions, literature references and yield (where available) is also stored. All the reactions can be easily browsed from within the synthesis planning software.

Development of algorithms and software modules for synthetic and retrosynthetic analysis

A number of algorithms and routines was developed that allow one to perform synthetic and retrosynthetic analyses (including detection of possible reactions and construction of the respective reagent structures).

The general procedure of retrosynthetic analysis can be formulated as follows:

- Preprocessing of the desired target structure (in particular, detection of the aromatic bonds)
- Calculation of the structure fingerprints
- Fingerprint-based prescreening of the reaction database
- Strict substructure search for the products of each generic reaction
- Retrosynthetic transformation (including transfer of target atoms not present in the generic reaction)
- Selection of valid reactions based on the atom valence constraints
- Generation of visual representation of the reaction

To expand the set of suitable reactions, the procedure is repeated using lower (less detailed) level of generic reaction representation using a number of heuristics.

For the fingerprint-based prescreening, the SQL queries of the following form are used:

```
SELECT s.id AS id, s.reaction_id AS reaction_id, LENGTH(s.structure_dump) AS dump_len,
s.structure_dump AS structure_dump FROM structure AS s, reaction AS r WHERE
r.reaction_type=_rtype_ AND r.id=s.reaction_id  AND s.structure_type=1 AND
s.fingerprint1&_fp1_=s.fingerprint1 AND s.fingerprint2&_fp2_=s.fingerprint2 AND
s.fingerprint3&_fp3_=s.fingerprint3 AND s.fingerprint4&_fp4_=s.fingerprint4 AND
s.fingerprint5&_fp5_=s.fingerprint5 AND s.fingerprint6&_fp6_=s.fingerprint6 AND
s.fingerprint7&_fp7_=s.fingerprint7 AND s.fingerprint8&_fp8_=s.fingerprint8
```

where `_rtype_` is the type of reaction (210 – RT, 215 – RC, 221 – ERC_S, 222 – ERC_T, 230 – ERC_M); `_fp1_`…`_fp8_` are the 1st…8th of the 64-bit fragments of the target structure fingerprint.

The procedure uses the substructure search and molecular graph manipulation routines developed in the course of the project.

The *algorithm of precursor structure construction* was refined to avoid generation of invalid reactions. In particular, for multi-step reactions present in a database, the reactants of a generalized transformation may contain atoms that do not have a corresponding atom in a transformation product. As a result, bonds not marked as broken in the transformation could be added to the precursor structure.

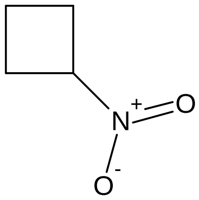
In the current version, the precursor construction algorithm works as follows:
Input: transformation reaction and the array of correspondence between transformation product atoms and target structure atoms.

1) Find mapping of transformation product to the target structure. Assign atom labels from the transformation product to the corresponding atoms in the target structure.
2) For each transformation reactant *rmol*:
    2.0) Create a copy of a target structure *tmol*.
    2.1) In *tmol*, remove the bonds marked as broken in the reaction. As a result, *tmol* may be split into several connected components. Replace *tmol* with the component containing atoms with the same labels as in the current reactant *rmol*.
    2.2) Note the correspondence between *rmol* and *tmol* atoms having the same labels. Then attempt to map the free atoms in *rmol* (having no match in *tmol*) to the free atoms in *tmol* (having no match in *rmol*), avoiding collisions in the adjacency matrices of *tmol* and *rmol*.
    2.3) In an endless loop, add the clones of free *rmol* atoms to *tmol*. Atom can be added if *tmol* contains an atom corresponding to its neighbor in *rmol*. In other words, consider *ra* atom in *rmol* such that its label is not present in *tmol*. Let *ran* be an *rmol* atom connected to *ra* by bond of the type *rbt*. Let *tan* be a *tmol* atom having the same label as *ran*. Then add to *tmol* a new atom *ta* as clone of *ra* (same type and label) and connect *ta* and *tan* with the *rbt* type bond. If the bond *ra-ran* is not marked as broken while *tan* and *ran* were mapped to each other in step 2.1 or 2.2 (rather than 2.3), this means that the current transformation cannot be applied to the target structure because the reaction would be invalid. In this case abort the procedure.
    2.4) Add to *tmol* the missing bonds (i.e., bonds between *rmol* atoms that have no corresponding bond in *tmol*) and adjust the *tmol* bonds that are changed in the reaction.
    2.5) The resulting *tmol* structure is a precursor of a target structure corresponding to the current transformation reactant.
3) If some atoms of the target structure are not labeled, they are assigned new labels not present in a transformation. This eliminates the need to analyze atom-to-atom correspondence for the resulting reaction in order to construct the ERC_M transformation.

Each proposed precursor compound can, in turn, serve as a target compound for the next step of the retrosynthetic analysis. In the synthesis planning software, it is selected for analysis by a mouse click.

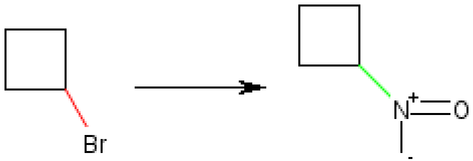
Let us present a few examples illustrating the operation of the synthesis planning function. (In the following reaction schemes, bonds marked in red are broken, bonds marked in green are created, and bonds marked in blue are changed in the course of a reaction).

1. Nitrocyclobutane

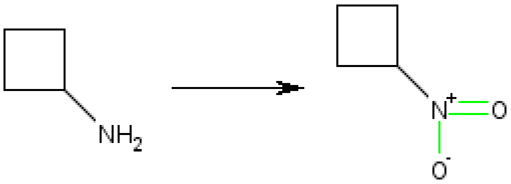


Several reasonable approaches to this compound are proposed by the software. Among them:

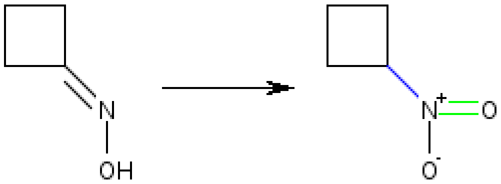
Substitution of bromine



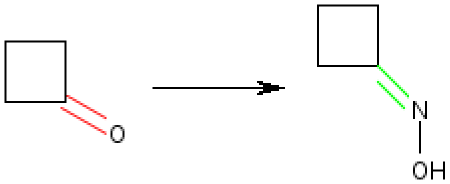
Oxidation of amine



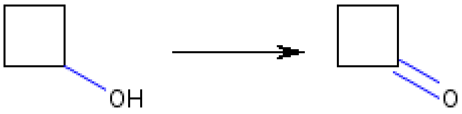
Oxidation of oxime



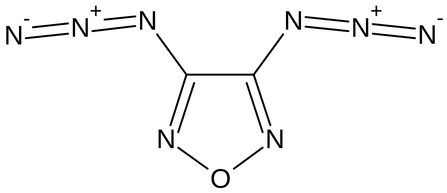
If the oxime is selected as precursor, the route from the cyclobutanone is suggested:



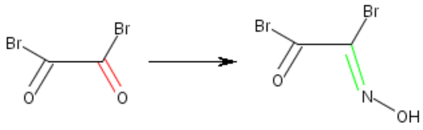
In turn, cyclobutanone may be obtained by cyclobutanol oxidation:

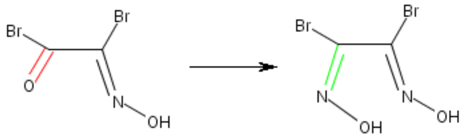


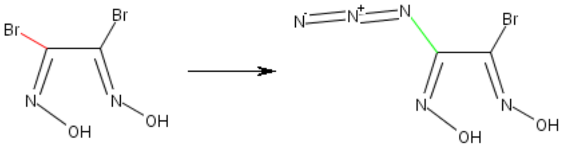
2. Diazidofurazan

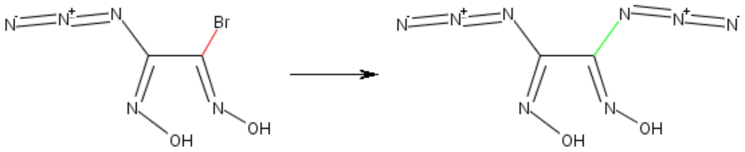


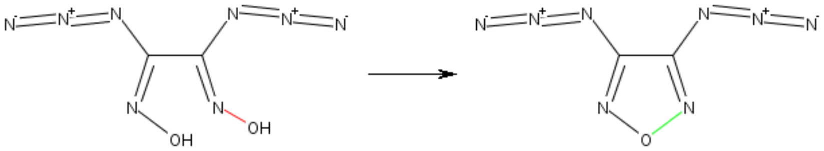
Several reasonable routes are identified, for example:


(oxime formation)

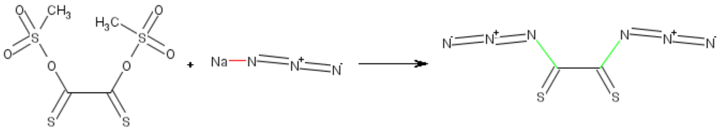

(oxime formation)

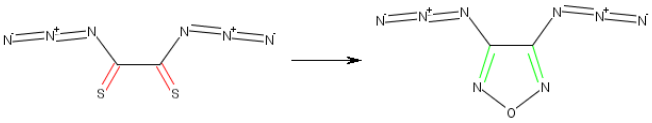

(bromine substitution by azide)


(bromine substitution by azide)


(ring closure)

Another route:


(mesylate substitution by azide)


(ring closure)

Thus, the current version of the software can be employed to suggest reasonable synthetic routes to compounds of interest.

Development of approaches to assessing the plausibility of proposed reactions and reaction paths

In most non-trivial cases, many different precursors and synthetic paths are possible for a given target compound. Some of them are more realistic than the others. Thus, the development of techniques for the quantitative assessment of plausibility of potential organic reactions proposed by the synthetic and retrosynthetic analysis routines was required.

Two approaches to such assessment were proposed:

- Analysis of structural similarity of proposed reaction to known successful reactions
- Statistical analysis to model the patterns common to the successful reactions

The first approach is based on the calculation of *similarity indexes of structural reaction fingerprints*. It can be summarized as follows.

Suppose that a given reaction ("precursor structures → target structure") was constructed from a certain generalized transformation (belonging to the ERC_M, ERC_T, ERC_S, RC or RT level). For a proposed reaction, the respective generalized ERC_M transformation is computed (reaction *R_1*), and its structural reaction fingerprint *FP_1* is formed. The structural reaction fingerprint (SRFP) is the bitwise-OR of the fingerprints of the reactants and the bit-rotated fingerprint of a reaction product. For generalized transformations, the corresponding starting and target structure fragments serve as 'transformation reactant' and 'transformation product', respectively. In contrast to the use of structural reaction fingerprints in some other systems for "reactant or product" type of substructure search during reaction database retrieval, here the fingerprints of reaction products are rotated by one bit to differentiate them from reactants and enhance the relevance of plausibility assessment.

If the proposed reaction is formed from a generalized transformation of the ERC_M type (reaction *R_2*), its structural reaction fingerprint (*FP_2*) is computed directly. Then the measure of similarity between *R_1* and *R_2* is estimated by means of the Tanimoto similarity index for *FP_1* and *FP_2*.

$$Tan(FP\_1, FP\_2) = \frac{count(FP\_1 \& FP\_2)}{count(FP\_1 | FP\_2)}$$

where *count(X)* is the number of non-zero (set) bits in *X*.
The value of the Tanimoto index provides an estimate of similarity between proposed reaction and real reaction present in a database. Thus, it is expected to characterize the plausibility of a proposed reaction.

On the other hand, if the proposed reaction is formed from a generalized transformation of the ERC_T, ERC_S, RC or RT type, it is necessary to retrieve all related ERC_M level transformations from the database. For each such transformation (reaction *R_2i*), the structural reaction fingerprint (*FP_2i*) is formed and the Tanimoto similarity index between reactions *R_1* and *R_2i* is computed from the fingerprint values *FP_1* and *FP_2i*. Among them, the transformation *R_2* providing the highest index value is then selected. This transformation is most similar to the ERC_M transformation of a proposed reaction, and the respective Tanimoto index can be used as a measure of its plausibility.

The original approach to reaction plausibility estimation was based on the Tanimoto similarity index between the reaction fingerprints of the full ERC_M transformations for the reaction in question (constructed by the system) and the reaction found in a database. The testing indicates that it may involve relatively wide environment of the reaction site (especially in compounds having big non-saturated or aromatic fragments). However, in most cases the reactivity is affected primarily by the relatively close environment (up to a few bonds). On the other hand, ERC_M-based similarity sometimes may overlook the influence of alkyl substituents. Thus, we have developed several alternative techniques of plausibility estimation in order to select the best approach:

- Tanimoto similarity index between the reaction fingerprints of transformations involving the reaction core environment up to 4 bonds (N4-FpR)
- Tanimoto similarity index between the reaction fingerprints of transformations involving the reaction core environment up to 5 bonds (N5-FpR)
- Mean value of the Tanimoto similarity indices between the fingerprints of reactants and products of the transformations involving the reaction core environment up to 5 bonds (N5-AvgFpM)

Let us consider the algorithm of plausibility assessment using these three methods.
Suppose that a given reaction ("precursor structures → target structure") was constructed from a certain generalized transformation (belonging to the ERC_M, ERC_T, ERC_S, RC or RT level).

To identify the transformation involving the required environment of the reaction core up to the depth of D bonds (ERC_D, D=4 or 5), the respective RC-level generalized transformation (connected reaction core) is found and its environment in the actual (CR-level) reaction up to the distance of D bonds is determined. Such a procedure is performed for the reaction constructed by the system (Tr_0) and for all the CR-level reactions (Tr_i, i=1,2,...N) related to the starting generalized transformation.

In the first two methods (N4-FpR and N5-FpR), for each of these transformations the reaction fingerprints are constructed, namely, FP_0 for Tr_0 and FP_i for Tr_i (i=1,2,..N). The Tanimoto similarity indices are then computed between FP_0 and each of the FP_i fingerprints, and the highest value is taken as a plausibility estimate.

In the third approach (N5-AvgFpM), the fingerprints are constructed for all reactants and a product of transformation Tr_0 and for all reactants and a product of each transformation Tr_i. Since the proposed reaction Tr_0 and the database CR-level reactions Tr_i are based on the same transformation, the number of reactants in Tr_0 and Tr_i is also equal. Moreover, the first reactant of the proposed reaction is an analog (albeit remote) of the first reactant of the database reaction, the second Tr_0 reactant is an analog of the second Tr_i reactant, etc. Thus, for each reaction pair (Tr_0 and Tr_i) we can compute a set of Tanimoto similarity indices:

– Tanimoto index between fingerprints of Tr_0 product and Tr_i product
– Tanimoto index between fingerprints of first Tr_0 reactant and first Tr_i reactant
– Tanimoto index between fingerprints of second Tr_0 reactant and second Tr_i reactant
– etc., for all Tr_0 and Tr_i reactants

Then the arithmetic average of all the Tanimoto indices for a given reaction pair (Tr_0 and Tr_i) is computed that provides a measure of similarity between these reactions. As before, the plausibility of the proposed reaction is estimated by the highest value of similarity measure among all Tr_i reactions.

When all potential precursors are constructed, the list of corresponding proposed reactions is sorted and displayed in descending order of reaction plausibility. The testing shows that the new approach indeed provides more accurate and intuitively desirable picture of reaction plausibility. By clicking the *“PL=X%”* hotlink, one can inspect the ERC_M transformations of the proposed and reference reactions used in the estimation of plausibility.

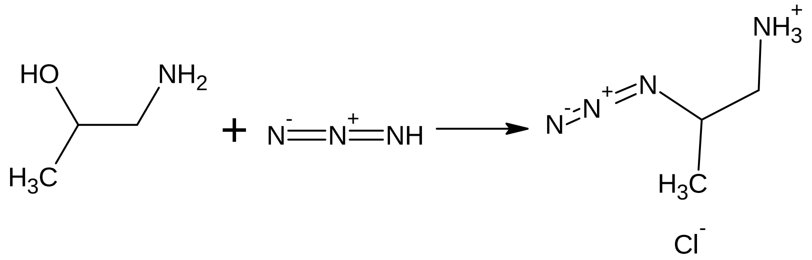
The extensive testing of the approach to reaction plausibility estimation based on the Tanimoto similarity index of reaction fingerprints reveals that such an estimation is sometimes too rough and fails to achieve the ranking of proposed reactions consistent with the experience of the chemist. An alternative approach is desirable that would enable the differentiation of ‘good’ vs. ‘bad’ reactions based on *statistical modeling of generalized patterns in a large set of experimental reactions*.

The major problem here is to obtain sufficiently representative body of ‘counterexamples’ for analysis by means of the statistical learning techniques – that is, a body of reasonable but failing reactions (called pseudoreactions). Such failed synthesis attempts are rarely reported in the literature and almost never compiled in the databases.

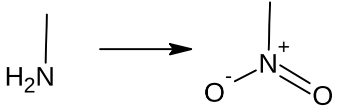
Thus, the first step was the construction of a set of pseudoreactions. More specifically, we define a pseudoreaction as a reaction derived by applying certain generalized transformation to a reagent different from its ‘native’ reagents (i.e., reagents participating in any of the original experimental reactions present in a database). A pseudoreaction involves one reagent and one product. During the generation, a set of the experimental CR-level reactions is scanned. In each reaction, the first reagent is selected as a pseudoreaction reagent. The product of a pseudoreaction is built using a generalized transformation randomly selected from the set of transformations present in the database (except the transformations derived from the current CR-level reaction). The search for a random transformation starts at the ERC_M level; if no suitable transformation is found, the search is continued in succession at the ERC_T, ERC_S, RC and RT levels. The preliminary screening of transformations at a given level is performed using the structure fingerprints, yielding a set of candidate transformations. Then, a single transformation is selected from this set by means of a random number generator. If the selected transformation is related to the original CR-level reaction, it is removed from the candidate list, and a new selection is made. After that, for the selected candidate transformation, it is checked if a molecular graph of its largest reagent (in terms of the atom count) is a subgraph of a selected reagent structure. If a negative result is obtained, the transformation is removed and another candidate transformation is selected.

Then the atom-to-atom matching of the pseudoreaction reagent and the largest transformation reagent is found. The pseudoreaction product is constructed based on a copy of the pseudoreaction reagent. The bonds marked as broken in the transformation reagent are removed, and the largest (in terms of the atom count) connected component is selected. Then the new atoms and bonds are introduced, and the bonds marked as changed are modified.

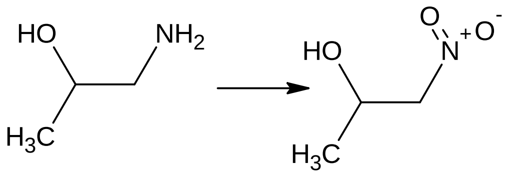
Let us present a few examples.
First original reaction is the conversion of hydroxy group in alcohols to azide .



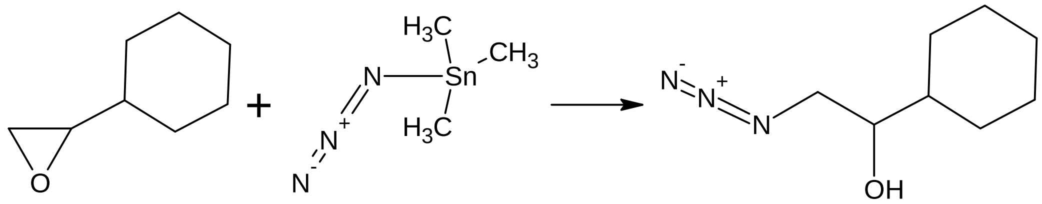
Using the transformation involving the oxidation of aliphatic amine to nitro group



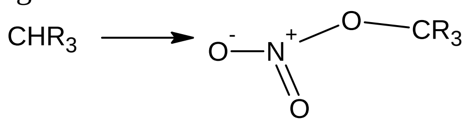
the following pseudoreaction of the same aminoalcohol is obtained:



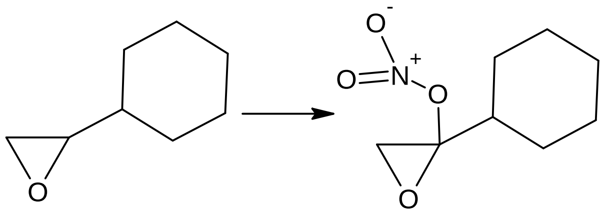
Second, for the reaction of stannylazide addition to epoxides



and the transformation involving nitrate formation from alkanes



one can obtain the following pseudoreaction:



For each original reaction and pseudoreaction, the fingerprints of the reaction center environment up to 4 and 5 atoms are constructed. From them, a uniform set of 512 attributes is derived such that each attribute represents a single bit value in the fingerprint. The analysis of these data by means of various statistical learning approaches should provide *statistical models for the estimation of plausibility of potential organic reactions*. Taking into account the complex non-linear nature of this property, we employed the Support Vector Machines (SVM) technique. This is a modern classification technique based on the implicit construction of the maximum-margin separating hyperplane in a certain transformed space (so-called Reproducing Kernel Hilbert Space). We used this approach to predict whether certain organic reaction belongs to the class of ‘good’ or ‘bad’ reactions. Since in the standard form the SVM technique provides only the ‘yes/no’ classification, the probability of class membership was estimated using Platt’s calibration procedure [51-55].

The SVM classifiers for estimating the likelihood of a reaction are built using the fingerprints for the real CR-level reactions stored in the database as well as the fingerprints for the pseudoreactions generated by means of the procedure described above.

At first, the classifiers were trained using the special data set containing 1011 known reactions and equal number of pseudoreactions that we treat as failing. Two classifiers are based on the descriptors representing the fingerprints of a reaction center environment up to the distance of 4 and 5 bonds, respectively. Their predictivity was estimated by means of the 10-fold cross-validation procedure. The accuracy of the first classifier (4-bond fingerprints) is 72.6% (sensitivity 74.0%, specificity 71.2%). The second classifier (5-bond fingerprints) has the accuracy 71.2% (sensitivity 71.8%, specificity 70.6%). Thus, the consideration of reaction center environment up to the distance of 4 bonds leads to slightly better classifier compared to the 5-bond environment.

In order to enhance the quality of the classifier, the set of pseudoreactions was extended. For each reactant of a CR-level reaction, an attempt was made to apply in the direct (synthetic) mode all the ERC_M-level transformations not related to the original reaction. If no pseudoreactions were built at the ERC_M level, we tried to apply ERC_T transformations, then ERC_S, etc. As a result, about 11000 pseudoreactions (negative examples) were built. The number of the positive examples remained 1011. Since the computational complexity of the SVM approach is approximately proportional to the cube of the data set size, the time required for the model construction was 2 orders of magnitude higher than in the previous case. Because of this, we were not able to find the classifiers with optimal parameter set and estimate their predictivity by means of cross-validation. However, even for the unoptimized model the accuracy of prediction on the balanced test set (containing equal number of the positive and negative examples) reaches 74.8% (sensitivity 83.3%, specificity 66.3%), surpassing the results for the smaller data set. Later we plan to employ the special techniques for handling large unbalanced data sets in the framework of the SVM approach that would provide substantial improvement in the classifier quality.

Detailed testing of different classifiers was performed in order to identify the most efficient ones. After construction of the reaction and pseudoreaction sets, 50 randomly selected reactions and 50 randomly selected pseudoreactions were set aside as a test set, and a training set was formed from other reactions and pseudoreactions.

The following cases were tested (in all four cases, the same reactions and pseudoreactions were contained in the test set):

- *fp4* – classifier based on the fingerprints of a 4-bond reaction site environment with equal number of reactions and pseudoreactions in the training set
- *fp4a* – classifier based on the fingerprints of a 4-bond reaction site environment with all available pseudoreactions in the training set
- *fp5* – classifier based on the fingerprints of a 5-bond reaction site environment with equal number of reactions and pseudoreactions in the training set
- *fp5a* – classifier based on the fingerprints of a 5-bond reaction site environment with all available pseudoreactions in the training set

The so-called ROC plots [56] were used to assess the quality of these binary classifiers. This approach involves the discrimination of the objects into two classes – positive and negative. For the reactions feasibility assessment, positive class corresponds to good reactions and negative class corresponds to bad reactions. In general, four cases are possible:

	Actual	
Estimated	**Positive**	**Negative**
Positive	*TP – true positive*	FP – false positive
Negative	FN – false negative	*TN – true negative*

Cases where positive and negative objects are recognized correctly are marked with italics. It can be seen that two types of errors are inherently possible:

- False negative – positive object incorrectly classified as negative (type I error)
- False positive – negative object incorrectly classified as positive (type II error)

Good classifier should achieve certain balance between these types of errors. To analyze it, two relative parameters are defined:

Sensitivity – fraction of correctly recognized positive objects $Se = \frac{TP}{TP + FN}$

Specificity – fraction of correctly recognized negative objects $Sp = \frac{TN}{TN + FP}$

Additionally, false positive rate $FPR = 1 - Sp$ is a fraction of incorrectly recognized negative objects.

ROC plot represents a plot of sensitivity *Se* versus false positive rate *FPR* obtained by varying certain classifier parameter such as the cut-off value. (The traditional name Receiver Operator Characteristic goes back to the use of such plots in the field of signal processing). In the ideal case, ROC plot should pass through the upper left corner where all objects are recognized correctly. Thus, the closer it is to the upper left corner, the better is the model. Conversely, less convex plots correspond to less efficient classifiers. Diagonal line ($Se = FPR$) represents a ‘useless’ classifier. Numerically, different ROC plots can be compared using the *Area Under Curve* (AUC) values. For ideal classifier, *AUC* is equal to unity.

Ideal classifier would have 100% sensitivity and specificity. However, in practice this goal is usually not attainable. For a given single-parameter classifier, it is impossible to increase sensitivity and specificity simultaneously, and certain balance or trade-off has to be reached that is commonly represented in terms of an optimal cut-off value. Several criteria have been proposed:

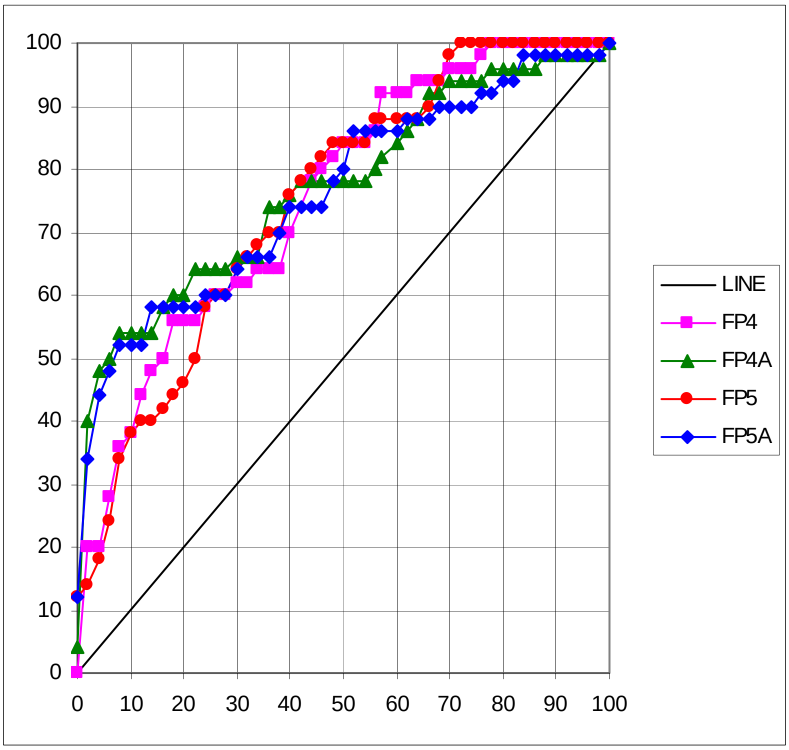
- Minimal required level of sensitivity or specificity, i.e., $Se > Se_{min}$ or $Sp > Sp_{min}$

- Maximal sum of sensitivity and specificity, i.e., $\max(Se+Sp)$
- Best balance between sensitivity and specificity, i.e., $\min|Se-Sp|$
- Some kind of weighting or prioritizing of sensitivity and specificity

Using this technique, the following results were obtained for the test set of 100 reactions and pseudoreactions:

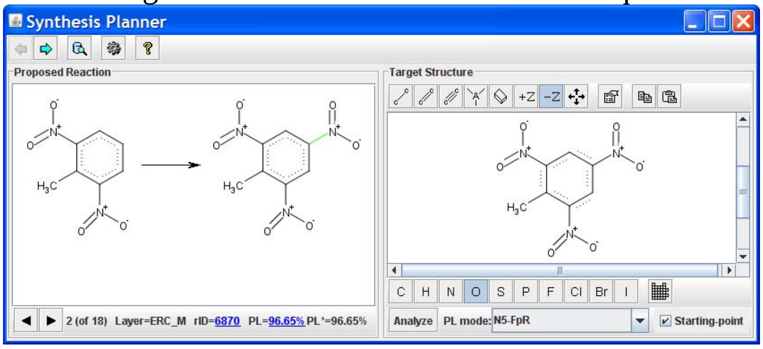
Classifier	Total Classification Accuracy	Area Under Curve
fp4	65%	0.75
fp4a	59%	0.77
fp5	67%	0.74
fp5a	56%	0.76

It can be seen that all four classifiers show rather good quality ($AUC > 0.75$). Their ROC plots are shown in the following figure. Additional research is needed in order to improve the quality of discrimination, optimize the cut-off parameters as well as to compare the SVM and Tanimoto-based classifiers.



All the steps of construction and usage of the SVM-based plausibility classifiers are now implemented within the synthesis planning software itself using LIBSVM library [57]. Thus, no external programs are needed.

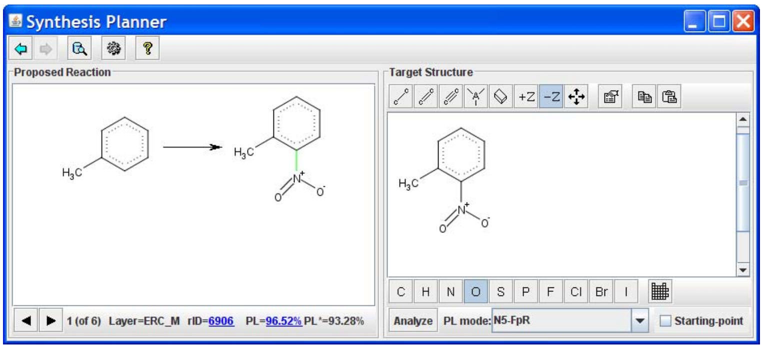
The proposed approach to *plausibility assessment for a reaction path* (a series of several sequential reactions leading to a target structure) is based on the multiplication of plausibility ratings of each individual reaction. This is conceptually similar to the calculation of joint probability of several independent random events as a product of probabilities of individual events (although direct analogy between probability value and plausibility rating is, of course, impossible). In the software, the resulting path plausibility estimate is displayed in the "*PL**" field. The final target structure in a series of reactions (i.e., the starting point of the retrosynthetic analysis) is specified by the "*Starting-point*" checkbox. As an example, the following figure shows the last and the first stage of the trinitrotoluene formation sequence.



2 (of 18) Layer=ERC_M rID=6870 PL=96.65% PL*=96.65%

PL mode: N5-FpR

☑ Starting-point



1 (of 6) Layer=ERC_M rID=6906 PL=96.52% PL*=93.28%

PL mode: N5-FpR

□ Starting-point

Development of algorithms and software modules supporting non-empirical approach to retrosynthetic analysis

To complement the empirical approach to computer synthesis, the non-empirical computer synthesis module was developed. It is intended to be used primarily as a 'fall-back' in cases where suitable synthetic approaches cannot be found by the empirical synthesis module using the available reaction data. In addition, it allows one to specify the rules for some special cases.

This module is based on the production-based expert system [58-61] for structure manipulation that was implemented in the synthesis planning software. A *production* is a certain "if-then" rule defined by a statement of a specialized production language. The syntax of this language and its interpreter were designed and developed by us. The goal was to ensure maximum expressiveness, ease of use and rich functionality.

The statement has the following format:

```
condition_predicate1(parameter_list), condition_predicate2(parameter_list), ...,
condition_predicateN(parameter_list)
-
action_predicate1(parameter_list), action_predicate2(parameter_list), ...,
action_predicateM(parameter_list)
.
```

The first part of the production (before dash) defines a set of conditions while the second part (after dash) defines a set of actions that are performed if the conditions are met.

In the current version of the interpreter the parameters can specify atoms (e.g., 1st C atom, 2nd C atom, 1st O atom, any or undefined atom) or numeric values. In future, they may be extended to support specification of bonds and/or molecular fragments.

- If the first character of a parameter is a digit, this parameter is interpreted as a non-negative integer number (e.g., 0, 1, 2, 16). Numeric parameters are used to specify a value of the atomic charge, bond order, etc.
- If the first character of a parameter is an upper-case letter, it defines a name of an atom of a given type, for example, C1, Cl2, O1. In this case, atom type is determined by the first and (if necessary) second alphabetical character while the remaining characters define some unique parameter key. The unique key is optional, and the parameter may contain only the atom type definition (e.g., O, Cl, Br).
- If the first character of a parameter is a lower-case letter, it simply defines a name of some atom.

In fact, atom names serve as variables. In other words, atom parameter can match any atom (if atom symbol is not specified in parameter) or any atom of a given type (if the symbol is specified). However, two parameters with different names cannot match the same atom.

Both condition and action predicates can be positive or negative. Negative predicates are specified by prepending the “**!**” (exclamation sign) character. Negative condition means that the respective condition should be false. Negative action is interpreted as a requirement to modify the structure in such a way that the respective predicate becomes false. For some predicates this may be impossible. For instance, the predicate `bond(a1, a2)` specifies the presence of bond of any type between atoms *a1* and *a2*. Thus, its negation `!bond(a1, a2)` is interpreted as the instruction to break the bond. On the other hand, the predicate `dbond(a1, a2)` specifies the presence of a double bond, and its negation `!dbond(a1, a2)` is ambiguous. It can be interpreted as an instruction to break the bond or change its order. Thus, such negative predicate is not allowed in the action list.

Currently the following predicates are implemented:

- `symbol(a, s)` – atom *a* has symbol *s*. Hidden predicate, cannot be specified explicitly. Such predicates are added to the condition list based on the analysis of the parameter names.
- `zp(a, z)` – atom *a* has positive or zero charge *z*. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.
- `zn(a, z)` – atom *a* has negative or zero charge -*z*. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.
- `radical(a, r)` – atom *a* has *r* unpaired electrons. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.
- `bond(a1, a2)` – atoms *a1* and *a2* are connected by a chemical bond of any type. This predicate can be specified both in the condition and action lists. Only negative predicate is allowed in the action list.
- `sbond(a1, a2)` – atoms *a1* and *a2* are connected by a single bond. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.
- `dbond(a1, a2)` – atoms *a1* and *a2* are connected by a double bond. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.
- `tbond(a1, a2)` – atoms *a1* and *a2* are connected by a triple bond. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.

- `abond(a1, a2)` – atoms *a1* and *a2* are connected by an aromatic bond. This predicate can be specified both in the condition and action lists. Negative predicate in the action list is not allowed.
- `incbond(a1, a2)` – increase the order of bond between atoms *a1* and *a2* (if no bond was present, connect atoms by a single bond). This predicate can be specified only in the action list and without negation.
- `decbond(a1, a2)` – decrease the order of bond between atoms *a1* and *a2* (if atoms were connected by a single bond, break this bond). This predicate can be specified only in the action list and without negation.
- `degree(a, n)` – atom *a* is connected to *n* other atoms (degree of vertex in a molecular graph). This predicate can be specified only in the condition list.
- `priority(n)` – production has priority *n*. This allows ranking of different productions based on their importance, likelihood, etc. This predicate can be specified only in the action list and without negation.

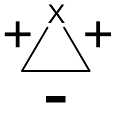
The production language supports comments similar to the C++ and Java languages. The comment is either started with the `//` characters and terminated at the end of line, or started with the `/*` and terminated with the `*/` characters. Comments can be used to add remarks and/or temporarily disable some parts of a statement.

The production language interpreter consists of a language parser and a production processing engine. The parser converts the statement from the text form to the internal representation. (If such a conversion is prevented by a syntax error, detailed error report is given.) The molecular structure and the production are passed to the processing engine. It builds all possible mappings of the atoms in a molecule and atomic parameters in the production. If the current mapping satisfies all the conditions of the production, all the specified actions are performed, yielding a new molecule. Thus, in the synthesis planning context, the output of the interpreter is a set of the potential precursor molecules obtained by applying the production to a target structure. (In other words, the production serves as a kind of a retrosynthetic rule). Duplicate precursor structures are removed using low-degeneracy 64-bit descriptor in a way similar to the duplicate removal in the empirical synthesis planning. If all mappings fail to satisfy the condition list or no mapping can be built (e.g., if the number of atomic parameters exceeds the number of atoms in the molecule), the output set would be empty.

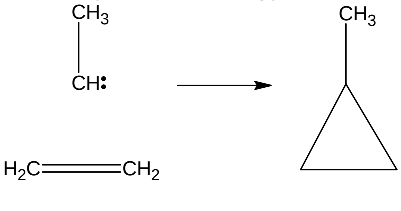
In the current version of the synthesis planning software, four production rules are specified that represent some of the most general cyclic bond redistribution topologies. For instance, the production

```
bond(a1,a2), bond(a1,a3), radical(a1,0) - incbond(a2,a3), decbond(a1,a2), decbond(a1, a3),
radical(a1, 1).
```

defines the following bond redistribution topology:



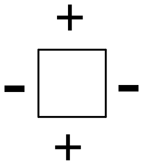
(Sign **+** means that in the course of a reaction the bond order is increased by 1; sign **–** means that it is decreased by 1; symbol **X** represents a special atomic center changing its valence by 2, valence of all other atoms is constant).
One example of a reaction represented by this topology is the carbene addition to alkenes:



Similarly, the production

```
bond(a1,a2), bond(a3,a4) - incbond(a1,a3), incbond(a2,a4), decbond(a1,a2), decbond(a3, a4).
```

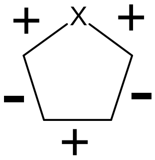
defines the 4-atom redistribution topology:



The production

```
bond(a1,a2), bond(a1,a3), bond(a4,a5), radical(a1,0) - incbond(a2,a4), incbond(a3,a5),
decbond(a1,a2), decbond(a1, a3), decbond(a4,a5), radical(a1,1).
```

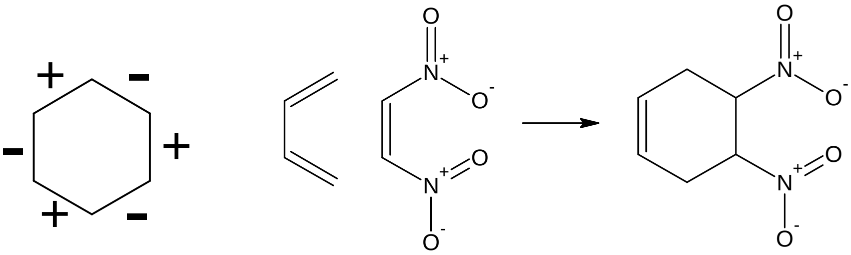
defines the 5-atom redistribution:



Finally, the production

```
bond(a1,a2), bond(a3,a4), bond(a5,a6) - decbond(a1,a2), decbond(a3,a4), decbond(a5,a6),
incbond(a1,a6), incbond(a2,a3), incbond(a4,a5).
```

defines the 6-atom redistribution topology exemplified by the Diels-Alder reaction, Cope rearrangement and a lot of similar reactions:



The applicability of the logical production language is not limited to the specification of the reaction schemes and synthetic strategies. For example, after implementing some additional predicates (e.g., related to the presence of certain functional groups, molecular fragments and/or environmental conditions) it can be used in construction of the expert system that would determine if one should use the protecting groups while performing a particular reaction.

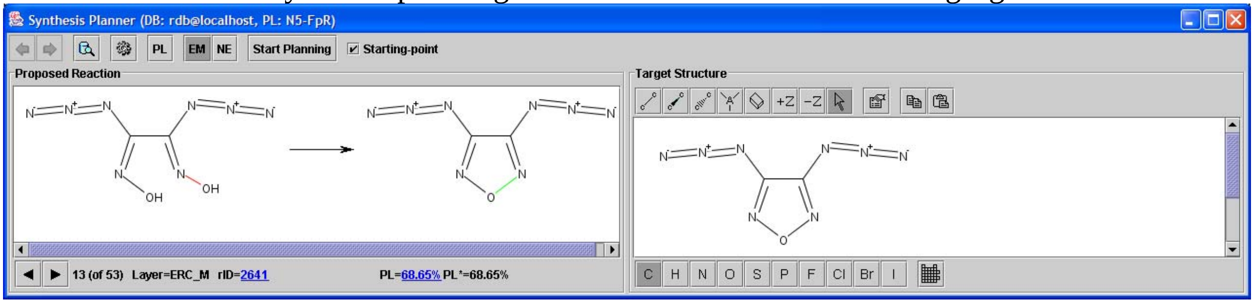
Development of the integrated, user-friendly synthesis planning system that implements all the required functionality

All the techniques described in the previous sections were implemented in a user-friendly computer-aided synthesis planning software. It is developed in the Java programming language and can be used on a wide variety of hardware and software platforms, including Windows, Linux and Mac OS. The information on chemical reactions is stored in the MySQL database. The following major features are supported:

- Experimental reaction data (including information on conditions, yields and literature references) can be imported from the standard RDF format files [5] and/or entered directly in the software.
- The hierarchy of generalized transformations is automatically constructed and adjusted as necessary.
- Reaction data (including information on conditions, yields and literature references, as well as atom-to-atom mapping and related generic reactions) can be browsed and edited.

- Target product structure can be easily entered using the graphical structure editor and/or copied and pasted via clipboard.
- The potential synthetic approaches and precursor structures are identified by means of retrosynthetic analysis based the available reaction database.
- For each proposed reaction, the information on similar experimental reactions (including conditions and literature references) is easily available.
- The proposed reactions are ranked by plausibility using reaction similarity index or statistical model.
- The statistical models based on the SVM (Support Vector Machine) technique can be constructed directly.
- Multi-step retrosynthetic analysis can be performed for complex syntheses, and the plausibility of multi-step reaction paths is estimated.
- Non-empirical synthesis planning based on a small number of reaction principles can be performed.

Main window of the synthesis planning software is shown in the following figure.


Synthesis Planner (DB: rdb@localhost, PL: N5-FpR)

☑ Starting-point

13 (of 53) Layer=ERC_M rID=2641 PL=68.65% PL*=68.65%

Results/Conclusions

The efforts in the framework of the present project are based primarily on the empirical approach to computer-aided organic synthesis planning. The underlying database of experimental reactions contains information on a substantial number of reactions. The generalized reaction patterns, rules and reaction principles are automatically extracted by means of data mining informational techniques in order to support inferring such transitions of chemical compounds that are not explicitly contained in databases.

In the course of a project, algorithms and software modules supporting all required functions of the synthesis planning software were developed. The MySQL database schema allows for efficient storage and fast retrieval of information concerning organic reactions, including chemical structures of reactants and reaction products, reaction conditions (temperature, solvents, additional reagents and/or catalysts, *etc.*), yield, and literature references, as well as the relations between generalized reactions of different levels. A set of flexible algorithms and software modules for manipulating molecular and reaction graphs provides fast solution of graph-theoretical problems occurring in the synthesis planning.

The concept of generic reactions representing certain common reaction patterns plays a central role in the computer-aided synthetic and retrosynthetic analysis. The following types of generic reactions may be considered to take into account different hierarchical levels of specificity and generalization. Concrete Reaction (CR) is a full chemical reaction with specific atom-to-atom mapping of reactants and products. All atoms that have at least one bond broken, formed or changed during the reaction are called reaction centers. Reaction Type (RT) is a generic reaction pattern including all the reaction centers. Reaction Core (RC) is a smallest connected graph including the RT pattern. To account for the influence of neighboring atoms, a number of Extended Reaction Cores (ERC) is considered. They are defined as RC pattern augmented by certain neighboring atoms. Generalized reaction transformation (TR) is a connected labeled graph derived from the set of the reaction center atoms. The procedures for automated detection of the reaction centers and recognition of the various levels of generic reactions from the raw chemical reactions reported in the synthetic literature were developed in the course of a project.

The database of organic reactions and transformations oriented to the synthesis of energetic compounds was created based on the literature data. It contains a significant number of reactions collected from literature, including a set of most important general-purpose organic reactions as well as specialized sets of reactions leading to furazan, furoxan, triazole, tetrazole, nitro and azido compounds. Total number of raw reactions in the database is 821, number of concrete reactions (one per product) is 840. From them, a total number of 2596 generalized transformations were derived at different levels of generalization. For each reaction, the information on reaction conditions, literature references and yield (where available) is also stored. All the reactions can be easily browsed from within the synthesis planning software.

The algorithms and routines for synthetic and retrosynthetic analyses were developed. They support detection of possible reactions, construction of the respective reagent structures, checking of the valence constraints and generation of visual representation of the reaction. Each proposed precursor compound can, in turn, serve as a target compound for the next step of the retrosynthetic analysis. The testing sows that the software can be employed to suggest reasonable synthetic routes to compounds of interest.

In most non-trivial cases, many different precursors and synthetic paths are possible for a given target compound. Some of them are more realistic than the others, and the techniques for the quantitative assessment of plausibility of potential organic reactions and reaction paths proposed by the synthetic and retrosynthetic analysis routines are required. Substantial effort was devoted to this problem, and two approaches to such assessment were proposed and implemented in the synthesis planning software. The first approach involves the analysis of structural similarity of proposed reaction to known successful reactions using the Tanimoto similarity indexes of their structural reaction fingerprints. Another approach is based on the statistical modeling of the patterns common to the successful reactions in contrast to related reasonable but failing reactions (pseudoreactions). The classification models are constructed by means of the Support Vector Machines (SVM) technique based on the implicit construction of the maximum-margin separating hyperplane in a certain transformed space. The testing shows that this approach is quite promising, although additional refinement is required.

To complement the empirical approach to computer synthesis, the non-empirical computer synthesis module was developed. It is intended primarily as a ‘fall-back’ for the cases where suitable synthetic approaches cannot be found by the empirical synthesis module using the available reaction data. This module is based on the production-based expert system for structure manipulation that was implemented in the synthesis planning software. A production is a certain “if-then” rule defined by a statement of a specialized production language. In the current version of the synthesis planning software, four production rules are specified that represent some of the most general cyclic bond redistribution topologies (covering, among others, the carbene addition and Diels-Alder reactions). This logical production language can also be used in construction of the expert systems that would identify optimal synthetic strategy or determine the necessity of the protecting groups while performing a particular reaction.

The powerful and user-friendly computer-aided synthesis planning software created in the course of a project provides all the features required for synthetic design. It is developed in the Java programming language and can be used on a wide variety of hardware and software platforms, including Windows, Linux and Mac OS. Among its major features are: import and direct entry of experimental reaction data (including information on conditions, yields and literature references); automatic construction of the hierarchy of generalized transformations; browsing and editing of the reaction data; easy specification of the target product structure; determination of the potential synthetic approaches and precursor structures by means of retrosynthetic analysis based the available reaction database; browsing of information on similar experimental reactions (including conditions and literature references) for each proposed reaction; ranking of plausibility of the proposed reactions using reaction similarity index or statistical model; planning and plausibility assessment of the multi-step reaction paths; non-empirical synthesis planning based on a small number of reaction principles.

Thus, in the course of a project, a substantial progress in the field of computer-aided synthesis planning was made. Major theoretical problems were solved, and efficient algorithms and software modules supporting all steps of the planning process were developed and implemented in the user-friendly cross-platform computer-aided synthesis planning software. The testing shows that it can be used to suggest reasonable synthetic routes to compounds of interest.

Future Work Recommended

In our opinion, further development of the computer-aided synthesis planning system should pursue the following general directions:

- The extension and refinement of the reaction and transformation database in order to introduce additional types of general-purpose organic reactions and additional reactions relevant for specific classes of compounds, as well as additional data concerning the reaction conditions.
- The development of advanced statistical techniques for the statistical evaluation of plausibility and probability of the reactions and reaction paths.
- Attempted qualitative prediction of reaction yield (low, moderate, high).
- The adaptation of the neural-network modules for the prediction of a number of relevant physico-chemical properties of several classes of target compounds (e.g., boiling point, density, enthalpy of formation).
- Further development of non-empirical synthesis approaches oriented to the cage hydrocarbons.
- Development of a convenient interface supporting the manipulation and analysis of the retrosynthetic trees in order to identify the most promising approaches.

As a result of additional effort, the computer-aided synthesis planning software will allow the researcher to obtain synthesis proposals based on richer set of available techniques as well as quickly and easily estimate the relevant physico-chemical properties of target compounds.

References

1. Zefirov N.S. Approach to systematization and design of organic reactions. *Acc. Chem. Res.* 1987, 20, 237-243.
2. Martirosov A.K., Goncharenko D.I., Zatsepin V.M., Ivanchenko V.A. Application of Markush formulae in modern chemical information systems: methodology and software technology means. *Nauch.-Techn. Inf.* 2004, Ser. 2, N 4, P. 21-33. (Russ.)
3. James C. A., Weininger D., Delaney J. Fingerprints-Screening and Similarity. Daylight Theory Manual; Daylight Chemical InformationSystems Inc.: 1997; (http://www.daylight.com/dayhtml/doc/theory/theory.toc.html)
4. Dobrynin D.A. Algorithms of fragment search in molecular graphs for automated encoding of chemical structures in intelligent drug design systems. *Nauch.-Techn. Inf.* 2002, Ser. 2, N 6, P. 51-57. (Russ.)
5. MDL CTfile Formats. Molecular Design Limited Inc. 2002. (http://www.mdli.com/solutions/white_papers/ctfile_formats.jsp)
6. Balaban A.T. Unique Description of Chemical Structures Based on Hierarchically Ordered Extended Connectivities (HOC Procedures). I. Algorithm for Finding Graph Orbits and Canonical Numbering of Atoms. *J. Comp. Chem.* 1985, 6, 538-551.
7. McKay B. D. Practical graph isomorphism. *Congressus Numarantium*. 1981. 30. P.45-87.
8. Scott F. The graph isomorphism problem. TR 96-20, Dept. of Computer Science. University of Alberta. 1996.
9. Kabekode V.S. Refined vertex codes and vertex partitioning methodology for graph isomorphism testing. *IEEE Transactions on Systems, Man, and Cybernetics.* 1980, SMC-10, 10.
10. Fortin S. Graph isomorphism problem. *Technical Report 96-20,* University of Alberta, Edmonton, Alberta, Canada, 1996.
11. Oliveira M., Greve F. A New Refinement Procedure for Graph Isomorphism Algorithms. *Proceedings of GRACO 2005: 2nd Brazilian Symposium on Graphs, Algorithms and Combinatorics,* April 2005.
12. Angra dos Reis RJ. *Electronic Notes in Discrete Mathematics (ENDM),* 19, Elsevier Group.
13. Ullman J. R. An algorithm for subgraph isomorphism. *J. ACM*. 1976. 23, P.31-42.
14. Chen L., Nourse J.G., Christie B.D., Leland B.A., Grier D.L. Over 20 Years of Reaction Access Systems from MDL: A Novel Reaction Substructure Search Algorithm. *J. Chem. Inf. Comput. Sci.* 2002, 42, 1296-1310.
15. Chen L. Substructure and Maximal Common Substructure Searching. In *Computational Medicinal Chemistry and Drug Discovery*; Tollenaere J., Bultinck P., Winter H. D., Langenaeker W., Eds.; Marcel Dekker: New York, 2003.
16. Zhu Q., Yao J., Yuan S., Li F., Chen H., Cai W., Liao Q. Superstructure Searching Algorithm for Generic Reaction Retrieval. *J. Chem. Inf. Model.* 2005, 45, 1214-1222.
17. Raymond J.W., Willett P. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 2002, 16, 521-533.
18. McGregor J.J., Willett P. Use of a maximum common subgraph algorithm in the automatic identification of ostensible bond changes occurring in chemical reactions. *J. Chem. Inf. Comput. Sci.*, 1981, 21, 137-140.
19. McGregor J.J. Backtrack Search Algorithms and the Maximal Common Subgraph Problem. *Software-Practice and Experience*, 1982, 12, 23-34.
20. Garcia G.C., Ruiz I.L., Gomez-Nieto M.A. Step-by-Step Calculation of All Maximum Common Substructures through a Constraint Satisfaction Based Algorithm. *J. Chem. Inf. Comput. Sci.* 2004, 44, 30-41.

21. Chent L., Robien W. MCSS: A New Algorithm for Perception of Maximal Common Substructures and Its Application to NMR Spectral Studies. 1. The Algorithm. *J. Chem. Inf. Comput. Sci.* 1992, 32, 501-506.
22. Balducci R., Pearlman R.S. Efficient Exact Solution of the Ring Perception Problem. *J. Chem. Inf. Comput. Sci.* 1994, 34, 822-831.
23. Downs G.M., Gillet V.J., Holliday J.D., Lynch M.F. Theoretical Aspects of Ring Perception and Development of the Extended Set of Smallest Rings Concept. *J. Chem. Inf. Comput. Sci.* 1989, 29, 187-206.
24. Fan B.T., Panaye A., Doucet J.-P., Barbu A. Ring Perception. A New Algorithm for Directly Finding the Smallest Set of Smallest Rings from a Connection Table. *J. Chem. Inf. Comput. Sci.* 1993, 33, 657-662.
25. Qian C., Fisanick W., Hartzler D.E., Chapman S.W. Enhanced Algorithm for Finding the Smallest Set of Smallest Rings. *J. Chem. Inf. Comput. Sci.* 1990, 30, 105-110.
26. Zamora A. An Algorithm for Finding the Smallest Set of Smallest Rings. *J. Chem. Inf. Comput. Sci.* 1976, 16, 40-43
27. Harary F. Graph Theory, Addison-Wesley, 1969.
28. Nechipurenko M.I., Popkov V.K., Mainagashev S.M. et al. Algorithms and programs for solving problems on graphs and networks. Nauka: Novosibirsk, 1990. (Russ.)
29. Barth A. Status and Future Developments of Reaction Databases and Online Retrieval Systems. *J. Chem. Inf Comput. Sci.* 1990, 30, 384-393.
30. Wang R., Wang L., Yuan Q., Luo S., Yao J., Yuan S., Zheng C., Brandt J. Construction of a generic reaction knowledge base by reaction data mining. *J. Mol. Graph. Mod.* 2001, 19, 427-433.
31. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 1. Introduction of New Concepts. *J. Chem. Inf. Comput. Sci.* 1986, 26, 205-212.
32. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 2. Classification of One-String Reactions Having an Even-Membered Cyclic Reaction Graph. *J. Chem. Inf. Comput. Sci.* 1986, 26, 212-223.
33. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 3. Classification of One-String Reactions Having an Odd-Membered Cyclic Reaction Graph. *J. Chem. Inf. Comput. Sci.* 1986, 26, 224-230.
34. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 4. Three-Nodal and Four-Nodal Subgraphs for a Systematic Characterization of Reactions. *J. Chem. Inf. Comput. Sci.* 1986, 26, 231-237.
35. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 5. Recombination of Reaction Strings in a Synthesis Space and Its Application to the Description of Synthetic Pathways. *J. Chem. Inf. Comput. Sci.* 1986, 26, 238-242.
36. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 6. Classification and Enumeration of Two-String Reactions with One Common Node. *J. Chem. Inf. Comput. Sci.* 1987, 27, 99-104.
37. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 7. Classification and Enumeration of Two-String Reactions with Two or More Common Nodes. *J. Chem. Inf. Comput. Sci.* 1987, 27, 104-110.
38. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 8. Synthesis Space Attached by a Charge Space and Three-Dimensional Imaginary Transition Structures with Charges. *J. Chem. Inf. Comput. Sci.* 1987, 27, 111-115.
39. Fujita S. Description of Organic Reactions Based on Imaginary Transition Structures. 9. Single-Access Perception of Rearrangement Reactions. *J. Chem. Inf. Comput. Sci.* 1987, 27, 115-120.
40. Fujita S. "Structure-Reaction Type" Paradigm in the Conventional Methods of Describing Organic Reactions and the Concept of Imaginary Transition Structures Overcoming This Paradigm. *J. Chem. Inf. Comput. Sci.* 1987, 227, 120-126.

41. Funatsu K., Endo T., Kotera N., Sasaki S.-I. Automatic recognition of reaction site in organic chemical reactions. *Tetrahedron Comp. Meth.* 1988, 1 (1), 53-69.
42. Lynch M.F., Willet P. The automatic detection of chemical reaction sites. *J. Chem. Inf. Comput. Sci.* 1978, 18, 154-159.
43. Zefirov N.S., Tratch S.S. Symbolic equations and their application to reaction design. *Anal. Chim. Acta.* 1990, 235, 115-134.
44. Tratch S.S., Zefirov N.S. A hierarchical classification scheme for chemical reactions. *J. Chem. Inf. Comput. Sci.* 1998, 38, 349-366.
45. Zefirov N.S., Baskin I.I., Palyulin V.A. SYMBEQ program and its application in computer-assisted reaction design. *J. Chem. Inf. Comput. Sci.* 1994, 34, 994-999.
46. Monson R.S. *Advanced organic synthesis*. Academic Press, London, 1971.
47. Ono N. *The nitro group in organic synthesis*. Wiley-VCH, New York, 2001.
48. Sheremetev A.B., Makhova N.N., Friedrichsen W. Monocyclic furazans and furoxans. *Adv. Heterocycl. Chem.* 2001, 78, 65-188.
49. Ostrovskiy V.A., Koldovskiy G.I. High-energetic 1,2,4-triazole derivativatives. *Russ. Khim. Zhurn.* 1997, 41, 73-83. (Russ.)
50. Petrov V.V., Bratilov S.I., Pantileenko S.V. Energetic tetrazoles. *Russ. Khim. Zhurn.* 1997, 41, 84-97. (Russ.)
51. Vapnik V. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
52. Cortes C., Vapnik V. Support-Vector Networks. *Machine Learning*, 1995, 20, 273-297.
53. Burges C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 1998, 2, 121-167.
54. Ivanciuc O. Applications of Support Vector Machines in Chemistry, in: *Reviews in Computational Chemistry*. 2007, 23, 291-400.
55. Platt J.C. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, in: *Advances in Large Margin Classifiers*, ed. A.J.Smola, O.Bartlett, B.Schoelkopf, MIT Press, 1999.
56. Fawcett T. *ROC Graphs: Notes and Practical Considerations for Researchers*. Kluwer, Dordrecht, 2004.
57. Chang C.-C., Lin C.-J. LIBSVM – A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
58. Russell S.J., Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, 2002.
59. Luger G. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley, London, 2002.
60. Naylor C. *Build your own expert system*. Wiley, Chichester, 1987.
61. Aho A.V., Sethi R., Ullman J.D. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, London, 1986.