УДК 004.42 DOI: 10.18698/2541-8009-2017-7-120

АВТОМАТИЧЕСКИЙ ВЫВОД СХЕМЫ СИНТЕЗА ОРГАНИЧЕСКОГО СОЕДИНЕНИЯ НА ОСНОВЕ ЕГО СТРУКТУРНОЙ ФОРМУЛЫ

Р.В. Замков zamkov.roman@gmail.com

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Предложена реализация программного обеспечения для аналитической работы при решении задачи планирования синтеза веществ на основе структурной формулы. Данное программное обеспечение позволяет получить пути синтеза требуемого органического соединения на основе небольшой базы знаний. Рассмотрена реализация данного программного обеспечения с помощью языков программирования Prolog и Python. Входными данными является целевая структура вещества, записанная в строчной нотации SMILES. Выходными данными является набор реакций, необходимых для получения заданной структуры, результатом работы — модуль на языке Python, который можно использовать для автоматического вывода путей синтеза органических соединений.

Ключевые слова

Планирование синтеза, хемоинформатика, RDKit, Prolog, Open Babel, молекулярный граф, реакционный граф, SMILES, Python

Поступила в редакцию 29.05.2017

Введение. Разработка программного обеспечения (ПО) для поиска путей синтеза веществ с заданной структурной формулой осуществляется в основном внутри коммерческих компаний и используется для внутренних нужд, либо для коммерческой реализации. В настоящее время современной академической реализации такого ПО не существует.

Целью данной работы является реализация ПО, составляющего схему синтеза требуемого вещества на основе его структурной формулы. Предполагается, что такое ПО будет использоваться в качестве компонента сценариев (скриптов), разрабатываемых в рамках исследовательских проектов в области синтетической химии, и в качестве компонента ПО для построения и анализа реакционных графов.

Постановка задачи. В работе использованы следующие понятия:

- молекулярный граф — связный неориентированный граф, соответствующий структурной формуле вещества [1];
- реакционный граф — связный ориентированный граф, вершины которого являются молекулярными графами;
- химическая реакция — переход от вершины к вершине реакционного графа согласно связям и меткам;
- база знаний — это набор реакционных графов (в данном исследовании).

Рассмотрим задачу планирования синтеза и прогноза путей метаболизма биологических веществ на основе решения прямой и обратной задач. В зависимости от направления обхода реакционного графа, условно разделим задачи на прямую и обратную.

Прямая задача заключается в планировании синтеза вещества на основе заданной структурной формулы, в том числе для поиска экономичного способа синтеза. В общем случае это задача прохождения по всем возможным путям заданного реакционного графа с учетом всех меток ребер. Для этого необходимо создать программный продукт, автоматически предоставляющий схему синтеза набор реакций, основываясь на структуре требуемого продукта (молекулярном графе) и базе знаний о химических реакциях.

Для решения такой задачи необходимо задать связный реакционный граф с точками входа и выхода. Каждая из вершин заданного графа является вершиной для перехода к задаче о выборе оптимального пути на графе. В число меток ребер входят коэффициенты, отражающие стоимость или длительность данного этапа синтеза. На входе — структурная формула вещества, которую требуется получить, в виде строки SMILES [2]. SMILES — это система правил, позволяющая записать структурную формулу и состав молекулы однозначным образом с помощью строки символов ASCII. Эта система правил используется многими программами-редакторами структурных формул и является компактной и удобной, широко применяется в коммерческом и академическом ПО. На выходе — набор последовательно проведенных химических реакций (план синтеза), согласно которому можно получить соединение с заданной структурной формулой. В базе знаний — реакционный граф, составленный по заранее известным правилам (на основе учебной литературы или баз данных), либо полученный эмпирическим путем.

При решении прямой задачи нам известно конечное вещество и набор веществ в базе (метки) требуется пройти по графу справа налево и вывести все вершины, с которыми есть связь.

Обратная задача — поиск возможных реакций, в которые может вступить вещество с заданной структурной формулой.

Исходными данными в этом случае являются структурная формула исходного вещества в виде строки SMILES и база знаний о реакциях (реакционный граф). Тогда результатом программы должны быть набор реакций, в которые может вступить вещество с заданной структурной формулой. В базе знаний — реакционный граф, содержащий все доступное множество реакций. Очевидно, что такие реакции должны храниться в базе знаний по возможности в виде шаблонов и правил, что позволит описывать реакцию для веществ одного класса посредством одной записи в базе. На рис. 1 приведен конкретный пример реакционного графа с веществами, заданными в виде общих формул.

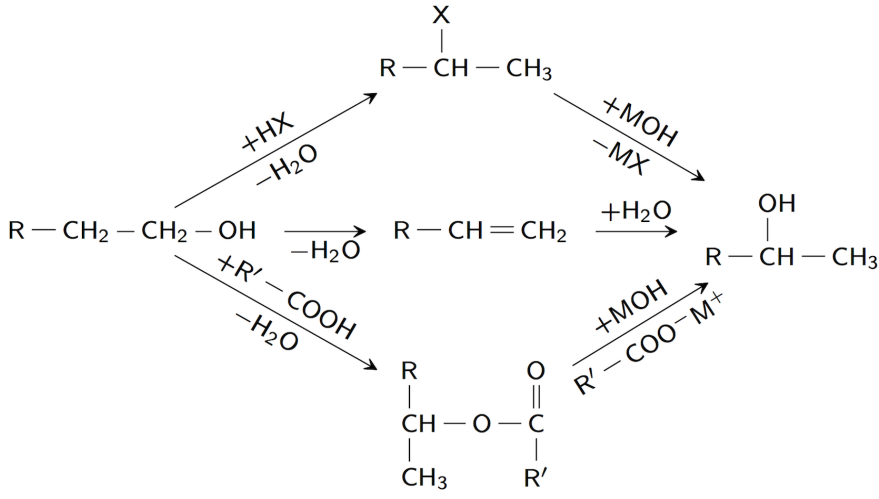


Рис. 1. Реакционный граф (схема синтеза) с веществами, заданными в виде общих формул

Стоит отметить, что вершин, с которых начинается обход, в реакционном графе может быть много, и какие из них будут использоваться, определяется задачей исследования. В свою очередь вершины, на которых будет завершен обход, соответствуют искомым веществам, и определяются содержанием базы знаний о реакциях.

Рассмотрим подробнее схему, представленную выше (см. рис. 1). Пусть задан реакционный граф с точками входа и выхода и требуется получить план синтеза вещества, либо пути метаболизма (реакции, в которые может вступить указанное вещество).

Прямая задача в этом случае будет заключаться в поиске вершины, соответствующей веществу, которое подано на вход, и выводу на экран всех возможных путей, ведущих к данной вершине с некоторым ограничением их количества. Например, в количестве пятнадцати, что позволило бы эксперту выбрать наиболее оптимальную схему из небольшого числа таких схем. Обратная задача в этом случае будет заключаться в прохождении графа от точки входа до точки выхода по заранее известному пути с выводом на всех этапах реакций и требуемых для их осуществления аддуктов.

Выбор средств обработки реакций. Для поиска путей решения задачи были использованы следующие средства: язык Prolog [3], пакеты классов и функций RDKit [4–6] и OpenBabel [7–9].

Язык Prolog. Prolog — это язык и система логического программирования, основанные на языке предикатов математической логики, представляющей собой декларативное подмножество Datalog. Поэтому с его помощью достаточно удобно описывать логические конструкции.

Преимущества использования Prolog.

1. Структурированность кода: он разделен на три группы (вершины, правила, направления перехода от вершины к вершине), что позволяет легко в нем разобраться.

2. Возможность удобного добавления новых элементов графа специалистом в предметной области.

Недостатки использования Prolog.

1. Требуется интерпретатор языка.

2. Чтобы сделать код универсальным (таким, чтобы была возможность работать с любыми реакциями и графами), требуется прописать очень много правил.

3. Низкая частота использования этого языка (как в сообществе программистов, так и в среде специалистов в предметной области).

Построение схемы синтеза на языке Prolog. Сначала следует ввод данных, последовательный вызов правил, по котором происходит переход от вершины к вершине (один из трех вариантов). Химическая реакция считается осуществимой, если известны две вершины, ребро между ними и указаны соответствующие метки ребер.

В качестве образца, с которым происходит сопоставление вводимых веществ, в программе Prolog используются правила, заданные с помощью конструкций языка Prolog (фактов, правил, сопоставлений с образцом).

Приведем пример записи вещества (факта) в Prolog :

```
radical('C').
reactant('Cl').
```

Запись правила выглядит так:

```
rxn([[R,'C','C','O','H'],['H',X]],[[R,'C',X,'C'],['H','O
','H']]) :- radical(R),reactant(X).
```

Пример вызова правила:

```
rxn(R,X).
```

В результате обращения к такому правилу получаем следующий результат :

```
[['C', 'C', 'C', 'O','H'], ['H', 'Cl']], [['C', 'C',
'Cl','C'],['H','O','H']];
```

Здесь R соответствует факту `radical('C')` и является фиксированной частью структуры вещества, X — факту `reactant('Cl').` и является фиксированной частью вещества-аддукта.

Данный пример относится к проведению первой реакции верхнего пути графа (см. рис. 1). В качестве значений R и X взяты вещества бутан (C_4H_{10}) и хлор (Cl_2) соответственно. Для описания работы требуется задать исходные вещества (факты на прологе), описать правило реакции и вызвать его. Вызов правила является аналогом вызова функции проведения реакции в других языках.

Пакет RDKit — набор инструментов для решения различных задач в области хемоинформатики. В нем есть все необходимое для описания и проведения

химических реакций с использованием нотации SMILES. Для реализации графа требуется сделать класс вершин, класс ребер и класс переходов по вершинам.

Преимущества использования пакета RDKit.

1. Поддержка распространенных языков программирования (Python, C++)[9].

2. Наличие большого числа инструментов для моделирования химических процессов, что позволяет сделать универсальный код.

3. Более глубокий анализ химических веществ средствами языка, так как в RDKit есть набор классов и функций, позволяющий работать с нужными для химических веществ объектами.

Недостатки использования пакета RDKit.

1. Требуется использовать дополнительно нотацию SMARTS, так как в RDKit реакцию можно описать только в этой нотации.

2. Имеются ограничения работы с некоторыми реакциями, например для описания ионного взаимодействия. Так, нотация SMILES не содержит конструкций для определения донорно-акцепторных связей.

В качестве примера, рассмотрим аналогичную предыдущей реакцию с помощью пакета RDKit. В нотации, применяемой в пакете RDKit и являющейся расширением SMILES, описание реакции примет следующий вид:

```
      Rxn                          =                          All-
Chem.ReactionFromSmarts('[*:1][C:2][C:3][O:4].[H,*:5]        >>
[*:1][C:2]([*:5])[C:3].[H2O:4]')
```

Применим данное правило, выполнив подстановку фрагментов CCCO и Cl:

```
      Rxn.RunReactants((Chem.MolFromSmiles('CCCO'),
Chem.MolFromSmiles('Cl')))
```

Работа программы с использованием RDKit. Начинают с ввода данных двух веществ в нотации SMILES, которые преобразуются в нотацию SMARTS, после чего вызывается правило, соответствующее реакции Rxn. Функция RunReactants проверяет, соответствуют ли введенные вещества шаблону левой части правила реакции Rxn, и если соответствует, то выводит правую часть согласно указанному правилу с поданными на вход веществами-аддуктами. На выходе получаем продукты в нотации SMARTS. Процесс повторяется для всех описанных правил. Реакция считается проведенной, если поданные на вход вещества соответствуют указанному в правиле шаблону SMARTS.

Пакет OpenBabel — это свободная химическая экспертная система, позволяющая анализировать, преобразовывать и хранить данные, при решении задач в области хемоинформатики.

Преимущества использования пакета.

1. Возможность работать с молекулами веществ как с отдельными объектами, имеющими конкретный набор необходимых для анализа свойств, описанных

в OpenBabel. В то время как в RDKit считается, что объект молекулы вещества уже задан нотацией SMILES, а в Prolog объект молекулы представляет собой непосредственный набор символов. Все необходимые свойства требуется описывать вручную и указывать дополнительные правила для их использования.

2. Гибкие возможности описания химических реакций, так как объект молекулы OpenBabel позволяет обращаться к каждому атому или связи вещества в отдельности, тем самым предоставляя возможность работы с молекулами как с конструктором.

3. Удобно создавать масштабируемое приложение.

Недостатком данного подхода является то, что реакции необходимо описывать вручную, то есть требуется написать функцию взаимодействия объектов молекул с логикой для каждого типа реакции, что требует подготовки базы знаний о реакциях специалистом в предметной области (химии) совместно с IT-специалистом.

Исходя из сказанного, выбрана реализация с помощью Python и OpenBabel, так как этот пакет позволяет сделать более универсальный и масштабируемый код средствами языка Python, а возможность описания реакции в ручном режиме позволяет описать весь необходимый набор процессов вплоть до взаимодействия конкретных атомов в веществе.

Особенности реализации. Реакционный граф всегда является ориентированным, что упрощает реализацию, так как это сокращает число возможных переходов от вершины к вершине. Обход такого графа зависит от вещества, путь синтеза которого хотим получить. Если поданное на вход вещество не соответствует ни одному шаблону вершины из базы знаний, программа сообщит что для данного вещества и указанного реакционного графа не удается найти ни один путь синтеза. База знаний задается ориентированным графом, и записывается в CSV файл согласно следующей структуре:

```
(Id; Vertex; Adducts; Connections; Products,Weight).
```

Реакция считается осуществимой, если имеются две вершины, соответствующие аддукту и продукту реакции, и между ними указано ребро (собственно реакция). Если вершина, соответствующая продукту, отсутствует, но указаны аддукт и условия проведения реакции, есть возможность найти нужный продукт с помощью класса реакций. Принцип работы такого класса представлен на рис. 2 и рис. 3 в виде UML-диаграммы деятельности. На выходе получается набор реакций (схема синтеза), по которым можно получить заданное вещество на основе загруженного реакционного графа.

Данный класс позволяет проводить процесс химической реакции, превращая строки SMILES в объекты химических молекул на уровне описанных в классе химических процессов, происходящих с молекулами. Для моделирования реакций распада получив продукты, можно снова подать их на вход и провести реакцию еще раз.

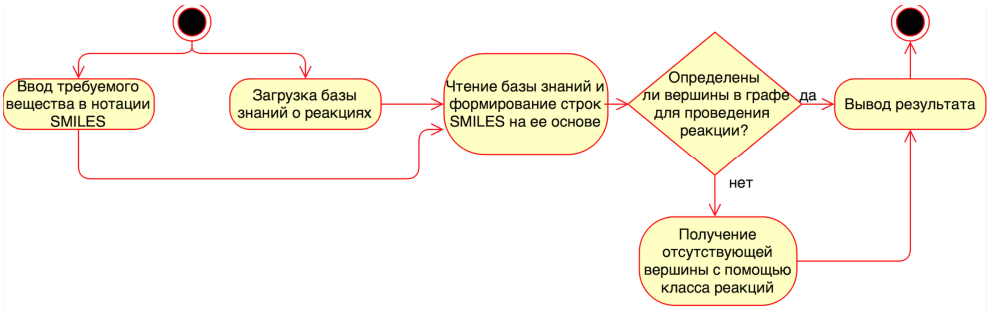


Рис. 2. Последовательность действий описываемого модуля в ходе работы

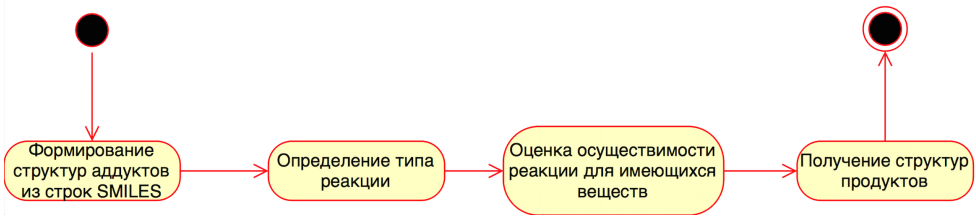


Рис. 3. Класс реакций: последовательность действий

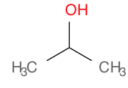
Примеры синтеза. Рассмотрим пример получения изопропилового спирта из пропанола-1, описанного с помощью реакционного графа (см. рис. 1).

Входные данные:

```
CCCO
```

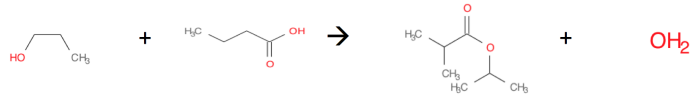
Результат:

```
V5 V4->V5
CC(C)OC(=O)C(C)C + [Li]O[H] -> CC(O)C + C(C)CC(=O)O[Li]
V4 V1->V4
CCCO + C(C)CC(=O)O[H] -> CC(C)OC(=O)C(C)C + [H]O[H]
cycle end
```

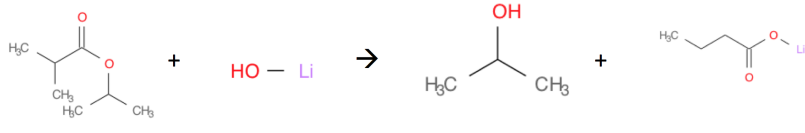
Требуемое вещество: 

Путь 1 :

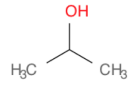
Реакция 1 V1->V4



Реакция 2 V4->V5:

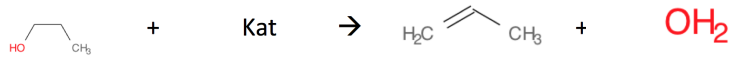


```
V5 V3->V5
CC=C + [H]O[H] -> CC(O)C + None
V3 V1->V3
CCCO + Kat -> CC=C + [H]O[H]
cycle end
```

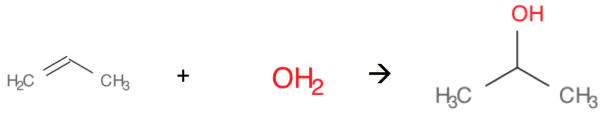
Требуемое вещество: 

Путь 2 :

Реакция 1 V1->V3

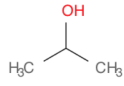


Реакция 2 V3->V5:



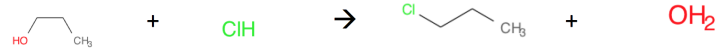
```
V5 V2->V5
C(Cl)CC + [Li]O[H] -> CC(O)C + [Li]Cl

V2 V1->V2
CCCO + [H]Cl -> C(Cl)CC + [H]O[H]
cycle end
```

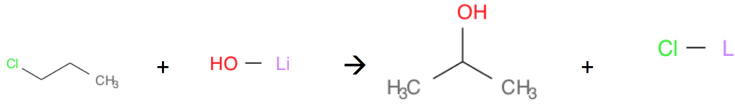
Требуемое вещество: 

Путь 3 :

Реакция 1 V1->V2



Реакция 2 V2->V5:



Данный вывод соответствует схеме (см. рис. 1) и свидетельствует о прохождении по верхнему, среднему и нижнему путям прохождения по графу.

Реакция `CCCO + [H]Cl -> C(Cl)CC + [H]O[H]` была рассмотрена нами в примерах работы с Prolog и RDKit. В традиционной записи уравнение имеет вид имеет вид: $C_3H_7OH + HCl \rightarrow ClC_3H_6 + H_2O$.

Верхний путь соответствует реакциям:

1) `CCCO + [H]Cl -> C(Cl)CC + [H]O[H]` (в традиционной записи $C_3H_7OH + HCl^{®} ClC_3H_6 + H_2O$);

2) `C(Cl)CC + [Li]O[H] -> CC(O)C + [Li]Cl`

(в традиционной записи $ClC_3H_6 + LiOH \rightarrow C_3H_8O + LiCl$).

Средний путь соответствует реакциям:

1) `CCCO + Kat -> CC=C + [H]O[H]` (в традиционной записи $C_3H_7OH \xrightarrow{t^o} C_3H_6 + H_2O$);

2) `CC=C + [H]O[H] -> CC(O)C + None`, (в традиционной записи $C_3H_6 + H_2O \rightarrow C_3H_8O$).

Нижний путь соответствует реакциям:

1) `CC(C)OC(=O)C(C)C + [Li]O[H] -> CC(O)C + C(C)CC(=O)O[Li]`

(в традиционной записи $C_3H_7OH + C_4O_2H_7 \rightarrow C_7O_2H_{13} + H_2O$);

2) `CCCO + C(C)CC(=O)O[H] -> CC(C)OC(=O)C(C)C + [H]O[H]`

(в традиционной записи $C_7O_2H_{13} + LiOH \rightarrow C_3H_8O + C_4O_2H_7Li$).

Данные вещества записаны в нотации SMILES и соответствуют всем веществам, указанным в графе (см. рис. 1), что подтверждает корректность работы программы. Таким образом на основе предложенного реакционного графа найдено три пути синтеза изопропилового спирта CC(O)C.

Выводы. В ходе работы рассмотрены варианты реализации решения задачи планирования синтеза. В результате выбран класс на языке Python, подходящий для автоматического вывода путей синтеза органических соединений при решении задачи планирования синтеза органических соединений. Достоинствами данного ПО являются высокая распространенность языка, эффективность экспертной системы OpenBabel, а также признание специалистами нотации SMILES и SMARTS.

Данный модуль можно использовать в составе ПО для планирования синтеза на основе ограниченной базы знаний. Например для решения более сложной задачи, требующей повторного получения путей синтеза на основе нескольких баз знаний, а также для решения задачи оптимизации расходов, при условии что в граф будут добавлены весовые коэффициенты меток реакций, имеющих смысл стоимости.

Литература

[1] Putta S., Eksterowicz J., Lemmen C., Stanton R. A novel subshape molecular descriptor. *Journal of Chemical Information and Computer Sciences*, 2003, vol. 43, no. 5, pp. 1623–1635.

[2] SMILES™. *Simplified molecular input line entry system*.
URL: http://www.daylight.com/smiles/ (дата обращения 21.12.2016).

[3] *SWI-Prolog downloads*. URL: http://www.swi-prolog.org/Download.html (дата обращения 12.02.2017).
[4] *RDKit. Open-source cheminformatics and machine learning*. URL: https://sourceforge.net/projects/rdkit/ (дата обращения 05.02.2017).
[5] *Molecular graph*. URL: https://goldbook.iupac.org/html/M/MT07069.html (дата обращения 14.03.2017).
[6] *API Documentation*. URL: http://openbabel.org/api/2.3/index.shtml (дата обращения 21.01.2017).
[7] *Open babel: the open source chemistry toolbox*. URL: http://openbabel.org/wiki/Main_Page (дата обращения 05.02.2017).
[8] *An overview of the RDKit*. URL: http://www.rdkit.org/docs/Overview.html#what-is-it (дата обращения 07.08.2016).
[9] *Python*. URL: http://openbabel.org/docs/current/UseTheLibrary/Python.html (дата обращения 04.02.2017).

Замков Роман Владимирович — студент кафедры «Теоретическая информатика и компьютерные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — А.В. Дубанов, канд. биол. наук, доцент кафедры «Теоретическая информатика и компьютерные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

AUTOMATIC CIRCUIT TERMINAL OF ORGANIC COMPOUND SYNTHESIS BASED ON ITS STRUCTURAL FORMULA

R.V. Zamkov zamkov.roman@gmail.com

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The study suggests software implementation for analytical work in solving the problem of planning substance synthesis based on the structural formula. This software makes it possible to get ways to synthesize a given organic compound using a small knowledge base. We consider this software implementation by means of two programming languages (Prolog and Python).The input data is the target substance structure, written in SMILES notation. The output data is the set of reactions necessary to obtain the given structure. The result of this work is a module in Python language, which can be used to automatically output ways of synthesizing organic compounds.

Keywords

Synthesis planning, chemoinformatics, RDKit, Prolog, OpenBabel, molecular graph, reaction graph, SMILES, Python

References

[1] Putta S., Eksterowicz J., Lemmen C., Stanton R. A novel subshape molecular descriptor. *Journal of Chemical Information and Computer Sciences*, 2003, vol. 43, no. 5, pp. 1623–1635.

[2] SMILES™. Simplified molecular input line entry system. URL: http://www.daylight.com/smiles/ (accessed 21 December 2016).

[3] SWI-Prolog downloads. Available at: http://www.swi-prolog.org/Download.html (accessed 12 February 2017).

[4] RDKit. Open-source cheminformatics and machine learning. Available at: https://sourceforge.net/projects/rdkit/ (accessed 05 February 2017).

[5] Molecular graph. Available at: https://goldbook.iupac.org/html/M/MT07069.html (accessed 14 March 2017).

[6] API Documentation. Available at: http://openbabel.org/api/2.3/index.shtml (accessed 21 January 2017).

[7] Open babel: the open source chemistry toolbox. Available at: http://openbabel.org/wiki/Main_Page (accessed 05 February 2017).

[8] An overview of the RDKit. Available at: http://www.rdkit.org/docs/Overview.html#what-is-it (accessed 07 August 2016).

[9] Python. Available at: http://openbabel.org/docs/current/UseTheLibrary/Python.html (accessed 04 February 2017).

Zamkov R.V. — student, Department of Theoretical Informatics and Computer Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — A.V. Dubanov, Cand. Sc. (Bio.), Assoc. Professor, Department of Theoretical Informatics and Computer Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.