Automatic generation of molecular zwitterionic forms with Ambit-Zwitterion

N. T. Kochev[1*], V. H. Paskaleva[1], N. Jeliazkova[2]

[1]*University of Plovdiv "Paisii Hilendarski", Department of Analytical Chemistry and Computer Chemistry,*
24 Tzar Asen Str., 4000 Plovdiv, Bulgaria
[2]*Ideaconsult Ltd, 4 A. Kanchev str., Sofia 1000, Bulgaria*

Received November 12, 2018; Revised December 10, 2018

We present a new open source tool for automatic generation of all zwitterionic forms of a given organic compound. Ambit-Zwitterion is an extension of Ambit-Tautomer software, part of the open source software platform Ambit and developed on top of a Chemistry Development Kit. All acidic centers in the target molecule (e.g. carboxylic groups, sulfonic groups, phosphoryl groups, etc.), as well as all basic centers (e.g., amino groups) are preliminarily identified. Ambit-Zwitterion implements a combinatorial algorithm for exhaustive generation of all zwitterions by making all combinations of *k* anions out of *n* acidic centers ($C_{k,n}$) and respectively all combinations of *k* cations out of *m* basic centers ($C_{k,m}$). Each combination from $C_{k,n}$ anionic set is combined with each combination from the $C_{k,m}$ cationic set to generate all zwitterionic forms containing *k* zwitterionic pairs. All zwitterions are obtained by iterating *k=1,2,…,min(n,m)*. Zwitterion generation algorithm is customized by a set of configuration options. Ambit-Zwitterion module is available for download as a Java library or as a command line application (https://doi.org/10.5281/zenodo.1481752). Software example usage and test results with Drug Bank structure set are presented.

Keywords: zwitterion, software, open-source, combinatorial, Ambit

INTRODUCTION

Zwitterions are neutral molecules that have separate positive and negative charge groups [1] in their structure. A lot of important biochemical and pharmaceutical compounds exist in zwitterionic forms. In physiological conditions, amino acids exist mainly as their zwitterions (see figure 1). Many zwitterionic structures have been used as drugs or prodrugs e.g. antibiotics, anti-HIV agents, diabetic drugs, etc. [2]. Zwitterion-derived materials have different organizational behaviors and they can be used as building blocks for nanostructured ionic materials [3]. The positive and negative charges of the zwitterions create specific ionic environment around proteins used as drugs and thus they can stabilize it with no affection of its bindings to the target [4]. They show potential as biocompatible coating for nanomaterials used in various drug formulations [5] and for passivating the surface of gold nanoparticles intended for *in vivo* applications [6]. Zwitterionic buffer additives are found to be useful in capillary zone electrophoresis for preventing the adsorption of proteins on the capillary wall and to improve the separation of proteins [7]. In molecular electronics, they can be used as molecular switches. Al-Kaysi *et al.* [8] reported the synthesis of a fluorescent molecular switch which is based on the interconversion between the fluorescent zwitterionic form and the non-fluorescent anionic state of a spiro-cyclic Meisenheimer complex of 1,3,5-trinitrobenzene.

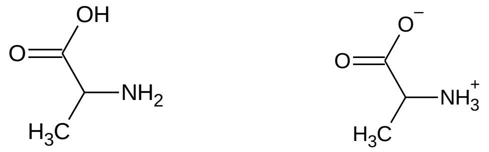


Figure 1. Alanine zwitterion

Computer-assisted drug design methodologies apply computer simulation to predict molecular properties as a function of the structure and are used to find the isomer responsible for bioactivity [2]. Adequate chemical structure representations are mandatory for the efficient utilization of chemoinformatics methodologies [9]. Molecular zwitterionic forms directly influence the topological structures and in this manner influence molecular descriptors and QSAR/QSPR models. Computer programs could be very helpful for the exploration of zwitterionic properties and predicting their activities. There are software tools for protonation and deprotonation of specific atomic types [10]. So far in the literature there are no reported open source software systems for exhaustive generation of zwitterions. The latter are especially needed for chemoinformatics handling of molecular structures with more zwitterionic centers. In this paper we present our newest development - Ambit-Zwitterion, a software tool for exhaustive generation of all zwitterionic forms of a target

* To whom all correspondence should be sent.
E-mail: nick@uni-plovdiv.net

chemical compound. We discuss its basic characteristics, usage examples and test results.

EXPERIMENTAL

Software characteristics

Ambit-Zwitterion is implemented in object-oriented programing language Java. It is an open source, OS independent software module distributed under LGPL license [11]. Ambit-Zwitterion is an extension of Ambit-Tautomer [12] (previously developed by us software tool) part of the chemoinformatics platform Ambit [13] where it is integrated as a separate module (ambit2-tautomers.zwitterion). Ambit integration allows usage of plenty of chemoinformatics functionalities from other Ambit modules [13] developed by our group, as well as utilities from external open-source resources. The source code of the java library extension (ambit2.tautomers.zwitterion) is available at http://ambit.sourceforge.net/. The source code of Ambit-Zwitterion command line application is available at http://ideaconsult.github.com/apps-ambit (folder tautomers-example) and an executable *.jar file can be downloaded from https://doi.org/10.5281/zenodo.1481752.

Software architecture

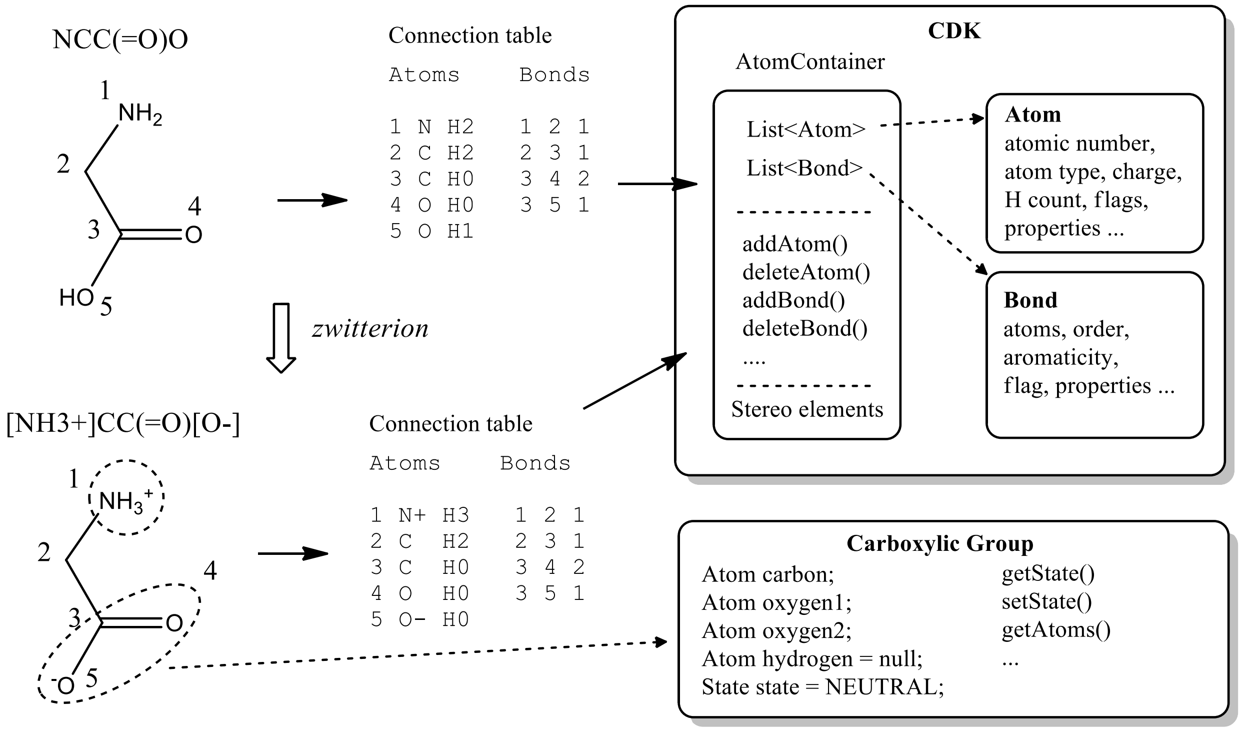
Ambit-Zwitterion implementation includes four basic components:

(1) Data input/output utilities. Ambit-Zwitterion supports basic formats for structure presentation: SMILES [14] and InChI [15] linear notations, CML chemical format, MOL/SDF file formats, CSV and TXT file formats. The software configuration can be done from command-line interface options. The variety of supported file formats allows easy integration of our tool with other software applications.

(2) Structural information management. Chemical structure representation in Ambit-Zwitterion is based on Java class AtomContainer which is part of the chemoinformatics library Chemistry Development Kit (CDK) [16]. AtomContainer implements the molecular connection table which is the foundation for all zwitterionic transformations (see figure 2).

(3) Chemical groups management. There are several dedicated java classes for handling the molecular groups corresponding to acidic and basic centers. For each group (acidic or basic) there are utilities for group detection and state changing e.g. NEUTRAL to ION and vice versa (see Carboxylic Group example in figure 2).

(4) Zwitterion generation. This is the most crucial software component. The exhaustive generation of all zwitterionic forms is based on the combinatorial algorithm described in the following section.



Connection table

```
Atoms      Bonds

1 N H2     1 2 1
2 C H2     2 3 1
3 C H0     3 4 2
4 O H0     3 5 1
5 O H1
```

Connection table

```
Atoms      Bonds

1 N+ H3    1 2 1
2 C  H2    2 3 1
3 C  H0    3 4 2
4 O  H0    3 5 1
5 O- H0
```

Carboxylic Group

```
Atom carbon;                   getState()
Atom oxygen1;                  setState()
Atom oxygen2;                  getAtoms()
Atom hydrogen = null;          ...
State state = NEUTRAL;
```

Figure 2. Chemical object management in Ambit-Zwitterion

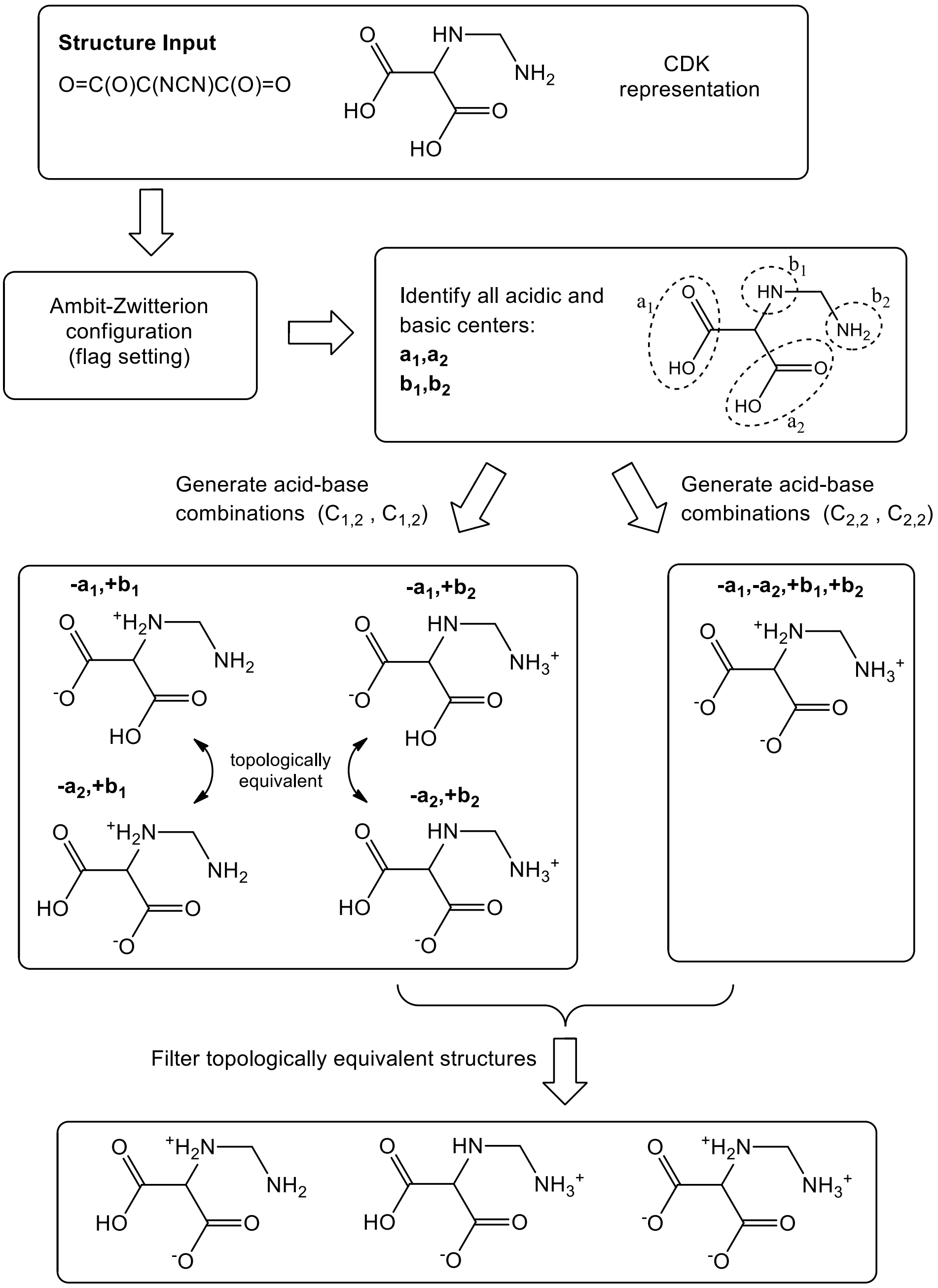

Structure Input
O=C(O)C(NCN)C(O)=O
CDK representation
Ambit-Zwitterion configuration (flag setting)
Identify all acidic and basic centers:
a1,a2
b1,b2
Generate acid-base combinations (C1,2 , C1,2)
Generate acid-base combinations (C2,2 , C2,2)
-a1,+b1
-a1,+b2
-a2,+b1
-a2,+b2
topologically equivalent
-a1,-a2,+b1,+b2
Filter topologically equivalent structures

Figure 3. Flow chart of the Ambit-Zwitterion basic generation sequence

Zwitterion generation algorithm

The basic workflow of Ambit-Zwitterion is summarized in figure 3. The molecule intput is taken from one of the popular molecular formats (SMILES, MOL etc.) and converted into the internal CDK representation. All acidic and basic centers are recognized (current version of Ambit-Zwitterion supports: carboxylic groups, sulfonic groups, sulfinic groups and phosphoryl groups).

After group detection, the initial sets of acidic centers *{a_1,a_2,..., a_n}* and basic centers *{b_1,b_2,...,b_m}* are obtained. The maximal number of simultaneous zwitterionic pairs, max_{zp}, in the target molecule is *min{n,m}*. All theoretically possible zwitterionic forms are obtained as a unification of all subsets of zwitterions containing *k* zwitterionic pairs simultaneously. The total number of generated zwitterions is:

$$Z = \sum_{k=1}^{max_{zp}} Z_k \quad , \qquad (1)$$

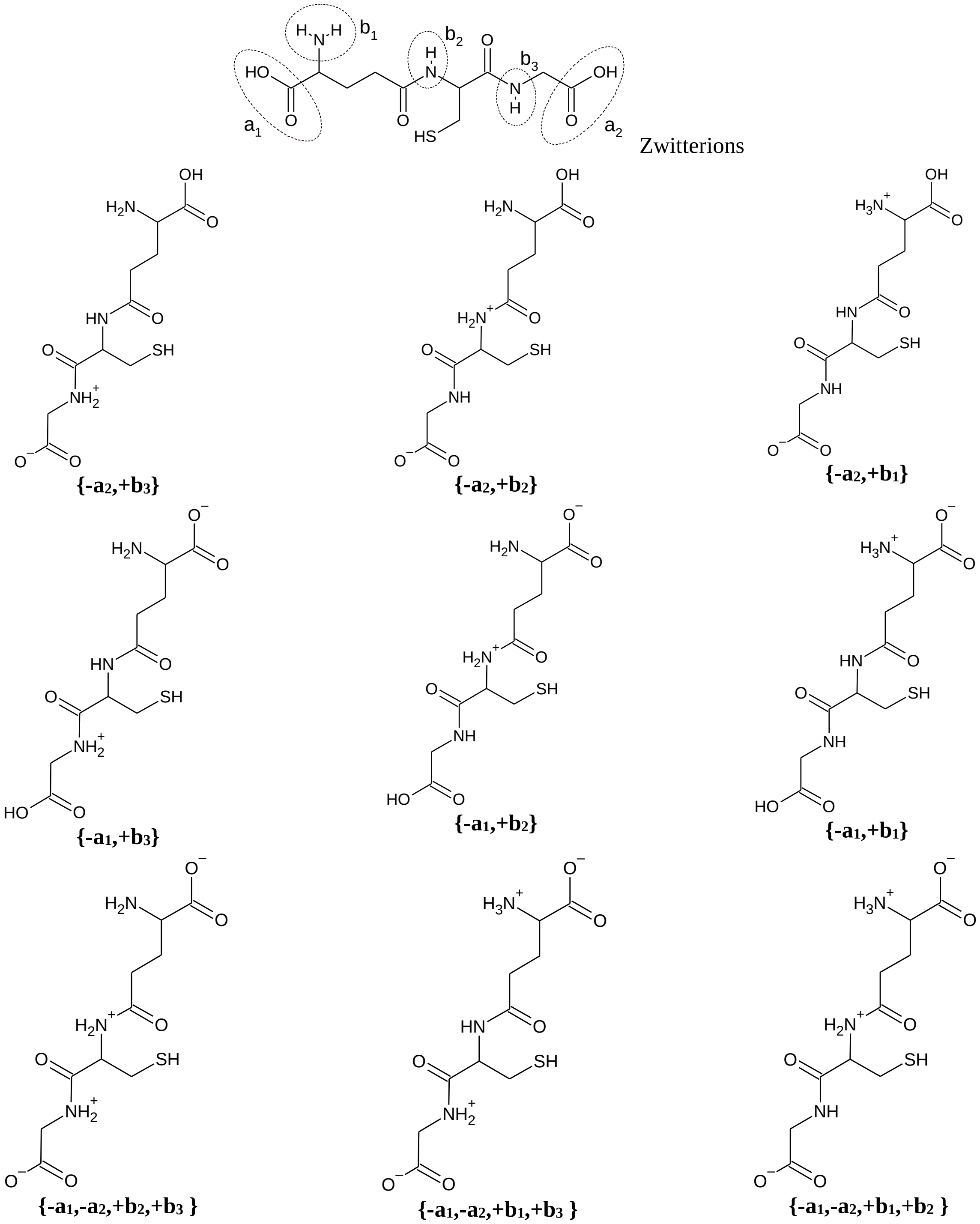
where Z_k is the number of zwitterions containing exactly *k* acid-base pairs. Ambit-Zwitterion implements a combinatorial algorithm for exhaustive generation by making all combinations of *k* anions out of *n* acidic centers ($C_{k,n}$) and respectively all combinations of *k* cations out of *m* basic centers ($C_{k,m}$). Each combination from $C_{k,n}$ anionic set is combined with each combination from $C_{k,m}$ cationic set to generate all zwitterionic forms containing *k* zwitterionic pairs. Thus the value of Z_k is obtained as:

$$Z_k = \left(\frac{n}{k}\right)\left(\frac{m}{k}\right) \qquad (2)$$

Figure 4 illustrates the combinatorial case: *n=2* and *m=3* for the molecule of glutathione. Initially sets {a_1,a_2 } and {b_1,b_2,b_3} are all in neutral state which corresponds to original molecule of glutathione with no zwitterions. The value of max_{zp} is *min(2,3) = 2*.

For *k=1*, six combinations are obtained:
{-a_1,+b_1} {-a_1,+b_2} {-a_1,+b_3}
{-a_2,+b_1} {-a_2,+b_2} {-a_2,+b_3}.
For *k=2*, three combinations are obtained:
{-a_1,-a_2,+b_1,+b_2 }
{-a_1,-a_2,+b_1,+b_3 }
{-a_1,-a_2,+b_2,+b_3 }

Totally, nine zwitterions are generated for the molecule of glutathione (figure 4).



{-a_2,+b_3} {-a_2,+b_2} {-a_2,+b_1}

{-a_1,+b_3} {-a_1,+b_2} {-a_1,+b_1}

{-a_1,-a_2,+b_2,+b_3 } {-a_1,-a_2,+b_1,+b_3 } {-a_1,-a_2,+b_1,+b_2 }

Figure 4. Zwitterion combinations for the molecule of glutathione ((2S)-2-amino-4-{[(1R)-1-[(carboxymethyl)carbamoyl]-2-sulfanylethyl]carbamoyl}butanoic acid)

As can be seen for the example input molecule from figure 3, topologically equivalent groups generate topologically equivalent zwitterions. For this purpose we have implemented a filtration option in Ambit-Zwitterion where structure duplications are removed by means of InChI keys [15] calculated with fixed H atoms option (the default InChI algorithm does not distinguish the zwitterionic forms).

Ambit-Zwitterion software version 1.0 is available as a command-line interface application with following options:

```
Zwitterion CLI
-c,--count                    Output count
                              statistics only
-f,--filter                   Filter duplicated
                              zwitterions (InChI
                              based)
-h,--help                     Zwitterion CLI

-i,--input <input>            Input molecule file
                              (*.smi, *.sdf, *.csv)
-m,--max_zwitterions <max>    Maximal number of
                              registered
                              zwitterions
-o,--output <output>          Output file name
                              (*.csv)
-p,--max_pairs <max>          Maximal number of
                              zwitterion pairs
-s,--smiles <smiles>          Input molecule smiles

-v,--verbose                  Verbose console
                              output
```

Example of Ambit-Zwitterion application for a molecule directly inputted as a SMILES notation from the command line is given as follows:

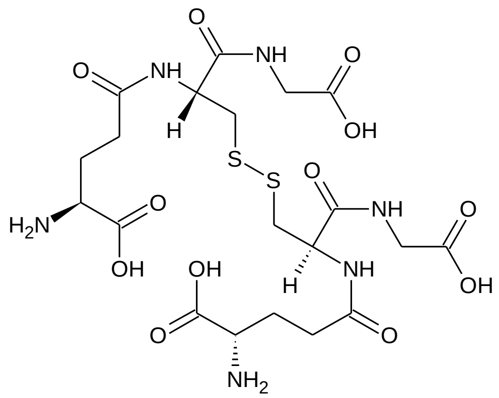
```
java -jar ambit-zwitterion.jar -s O=C(O)C(NCN)C(O)=O
-f


Input molecule: O=C(O)C(NCN)C(O)=O

Zwitterions:

C(N)[NH2+]C(C(=O)O)C(=O)[O-]

C([NH3+])NC(C(=O)O)C(=O)[O-]

C([NH3+])[NH2+]C(C(=O)[O-])C(=O)[O-]
```

Ambit-Zwitterion can be also applied in a batch mode for a set of molecules specified by means of -*i* option. Executable jar files and more examples are available at: https://doi.org/10.5281/zenodo.1481752.

RESULTS AND DISCUSSION

The molecular structure of oxiglutathione as well as zwitterion generation statistics obtained with and without filtration of topologically equivalent zwitterion structures are given in Figure 5. The total number of generated zwitterionic forms is 109. Figure 5 shows the number of structures containing one (Z1), two (Z2), three (Z3) and four (Z4) zwitterionic pairs of the type {acidic anion – basic cation}. As it can be seen, 12 zwitterionic structures are generated that contain one zwitterionic pair while most of the predicted theoretically possible zwitterions are obtained using combinations of two or three zwitterionic pairs.

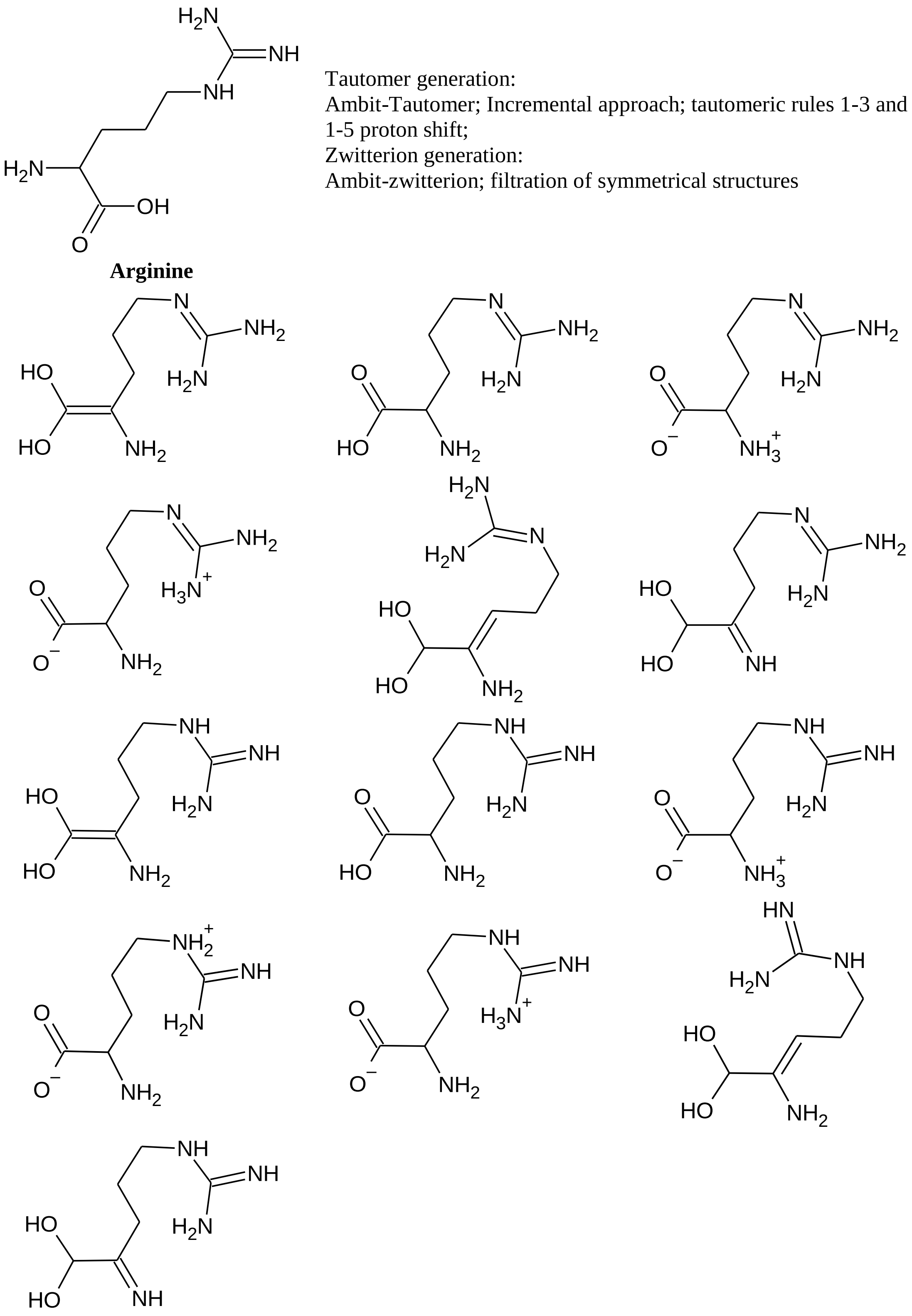
Another test result for the structure of arginine is given in figure 6. All arginine tautomers were generated using Ambit-Tautomer [12] software tool applying Incremental approach with tautomeric rules covering 1-3 and 1-5 proton shift.



Number of theoretical zwitterions:
Filter (no-filter)
109 (209)

Zwitterionic pairs	Number of zwitterions generated
Z1	12 (24)
Z2	48 (90)
Z3	40 (80)
Z4	9 (15)

Figure 5. Molecular structure of oxiglutathione and the theoretical number of its zwitterions and corresponding zwitterionic pairs



Tautomer generation:
Ambit-Tautomer; Incremental approach; tautomeric rules 1-3 and 1-5 proton shift;
Zwitterion generation:
Ambit-zwitterion; filtration of symmetrical structures

Arginine

Figure 6. Arginine molecule, generated tautomeric and zwitterionic forms

We have performed automatic tests generating all zwitterionic forms for a dataset of 6406 molecular structures (Drug Bank v.5.0.6 without salt containing substances). Computational time comparison was made for the zwitterion counting procedure with and without filtration of the topologically equivalent structures, as well as for the procedure of full generation of all theoretical zwitterions. The tests were made on a PC computer with RAM 8GB, Processor Intel® Core™ i5-82500U CPU @ 1.60GHz 1.80GHz, working on Windows 10, 64-bit operation system. The measured times, presented in table 1, also include the time for file reading and writing operations and SMIELS generation procedures.

Table 1. Zwitterion generation computational times for Drug Bank dataset

	Count (h:min:sec)	Full (h:min:sec)
Filter	0:04:40	0:09:45
No-filter	0:00:12	0:05:10

As it can be seen from table 1, all zwitterions (about 90 000 zwitterions for the entire Drug Bank set) were generated and counted for 12 sec. The actual storage on the output files took about 5 minutes that is mainly due to the SMILES generation (file operations time could be neglected). Accordingly, the needed time for filtration is about 4 minutes and 30 seconds and it is due to the computation of InChI keys used for topologically equivalent structures filtration.

Also, we have performed structural analysis of the obtained Drug Bank zwitterions summarized in figures 7, 8 and 9 (the shown statistics are with filtration of symmetrical molecules). Only 1451 molecular structures gave at least one zwitterion. Most of these molecules (36%) gave exactly one zwitterionic form and 27% showed theoretical possibility for two zwitterionic forms. Figure 7 shows the percentages of structures giving a number of zwitterions in the range 1-34 (i.e. statistical bins greater than or equal to 1%).

Figure 8 visualizes the percentages of structures that gave a number of zwitterions less than 1% (i.e. the histogram of rare zwitterionic counts).

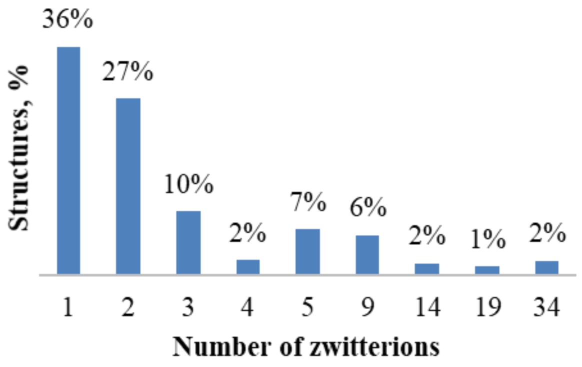


Figure 7. Distribution of the relative number (>1%) of structures generating from 1 to 34 zwitterionic forms

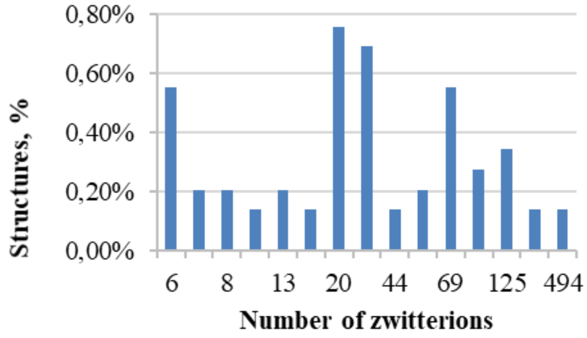


Figure 8. Distribution of the relative number (<1%) of structures generating a higher number of zwitterions (from 6 to 494+)

.

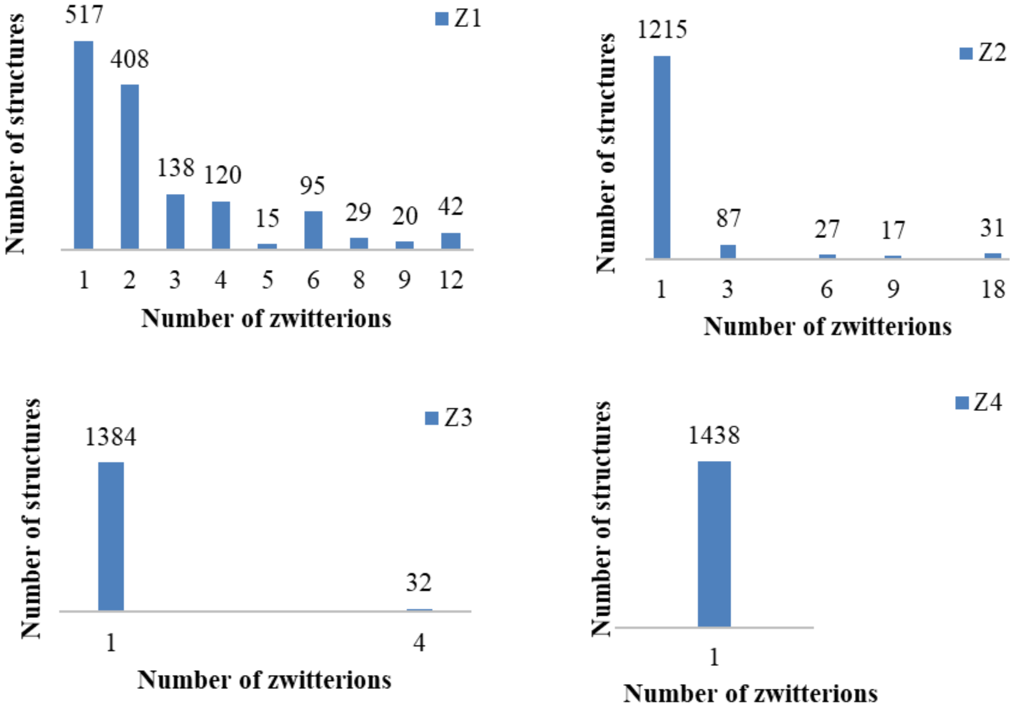


Figure 9. Distribution of generated zwitterions containing one (Z1), two (Z2), three (Z3) and four (Z4) zwitterionic pairs

Figure 9 shows the structure frequency distribution according to the number of zwitterionic pairs. Detailed test results are available at Zenodo repository at the following address: https://doi.org/10.5281/zenodo.1481752.

CONCLUSIONS

A new software tool (Ambit-Zwitterion) for exhaustive generation of all theoretically possible zwitterionic forms is developed. Software performance is demonstrated with various use cases and large-scale tests performed with Drag Bank molecules. Ambit-Zwitterion source code is available at http://ambit.sourceforge.net/ and the software can be easily integrated as part of bigger scientific workflows. Executable jar file with the latest version, additional examples and full test results are available at: https://doi.org/10.5281/zenodo.1481752.

Acknowledgement: *We thank Plovdiv University Scientific Fund (project MU17-HF-027) for supporting this scientific work.*

REFERENCES

1. https://goldbook.iupac.org/html/Z/Z06752.html, last accessed on 27.09.2018.
2. Z. Yang, Q. Li, G. Yang, *Future Med. Chem.*, **8**, 2245 (2016).
3. T. Ichikawa, *Polymer Journal*, **49**, 413 (2017).
4. A. Keefe, Sh. Jiang, *Nature Chemistry*, **4**, 59 (2012).
5. N. Hadjesfandiari, A. Parambath, Stealth coatings for nanoparticles: Polyethylene glycol alternatives, *Eng. Biomater. Drug Deliv. Syst.*, 345 (2018).
6. L. Knittel, P. Schuck, C. Ackerson, A. Sousa, *RSC Adv.*, **6**, 46350 (2016).
7. X. Fang, T. Zhu, V. Sun, *J. High Resolut. Chromatogr.*, **17**, 749 (1994).
8. R. Al-Kaysi, G. Guirado, E. Valente, *European J. Org. Chem.*, 2004 (16), 3408 (2004).
9. T. Engel, J. Gasteiger, Chemoinformatics: basic concepts and methods, Wiley-VCH Verlag GmbH, 2018.
10. N. O'Boyle, M. Banck, C. James, C. Morley, T. Vandermeersch, G. Hutchison, *J. Cheminform.*, **3**, 33 (2011).
11. LGPL: https://www.gnu.org/licenses/lgpl-3.0.en.html, last accessed on 27.09.2018
12. N. Kochev, V. Paskaleva, N. Jeliazkova, *Mol. Inform.*, **32**, 481 (2013).
13. N. Jeliazkova, V. Jeliazkov, *J. Cheminform.*, **3**, 18 (2011).
14. D. Weininger, *J. Chem. Inf. Comput. Sci.*, **28**, 31 (1988).
15. InChI: https://iupac.org/who-we-are/divisions/division-details/inchi/, last accessed on 27.09.2018.
16. E. Willighagen, J. Mayfield, J. Alvarsson, A. Berg, L. Carlsson, N. Jeliazkova, S. Kuhn, T. Pluskal, M. Rojas-Chertó, O. Spjuth, G. Torrance, Ch. Evelo, R. Guha, Ch. Steinbeck, *J. Cheminform.*, **9**, 33 (2017).