PolyMaS: A new software to generate high molecular weight polymer macromolecules from repeating structural units

Santiago A. Schustik[1), 2)] (ORCID ID: 0000-0003-1402-7055), **Fiorella Cravero**[1)] (0000-0002-9816-4147), **M. Jimena Martínez**[3)] (0000-0002-0443-5795), **Ignacio Ponzoni**[4), 5)] (0000-0002-6923-9592), **Mónica F. Díaz**[1), 6), *)] (0000-0002-6680-8067)

DOI: dx.doi.org/10.14314/polimery.2021.5.2

Abstract: The Polymer Maker SMILES-based (PolyMaS) software was used to generate linear macromolecules from the repeating structural units (SRU) of polymers without limiting their length and molar mass. The SRU input is stored in the SMILES code available on the Internet. PolyMaS makes head-tail junctions to the desired length of the macromolecule.

Keywords: computer design of macromolecules.

PolyMaS: Nowe oprogramowanie do generowania makrocząsteczek polimerów o dużej masie cząsteczkowej z powtarzalnych jednostek strukturalnych

Abstrakt: Oprogramowanie *Polymer Maker SMILES-based* (PolyMaS) zastosowano do generowania liniowych makrocząsteczek z powtarzalnych jednostek strukturalnych (SRU) polimerów, bez ograniczania ich długości i masy molowej. Dane wejściowe SRU są zapisane w dostępnym w Internecie kodzie SMILES. PolyMaS wykonuje połączenia głowa-ogon do żądanej długości makrocząsteczki.

Słowa kluczowe: projektowanie komputerowe makrocząsteczek.

The great advance in the field of polymeric materials, particularly in the relationship between the molecular structure and its properties, led to the incorporation of *in silico* tests [1–5]. They are virtual tests, carried out to know the polymeric material properties in the early stages of design, before being synthesized [6–8]. In this sense, the motivation to develop *in silico* techniques is related to the significant saving of resources that must be used, considering the economic point of view as well as the time for research and development [9]. One way of carrying out these virtual tests is to use quantitative structure-property relationship (QSPR) models [10]. Traditional synthetic representations, such as monomers, dimers, and trimers, have been used to generate these models. However, to obtain models which are closer to actual polymeric molecules, computational representations of very long polymer chains are necessary, and these representations can be obtained by the generation of macromolecules from the structural repetitive unit (SRU). Moreover, in the case of polymeric material databases, in which polydispersity is present, a more realistic representation should include several chain lengths [10].

Although, there are some proprietary software tools that offer generation of macromolecules from SRU for polymers like HyperChem® [11], Amsterdam Modeling Suite [12], QuantumATK [13] *etc*., they are not efficient or useful enough to our work. HyperChem®, which provides an option of polymerization from a monomer (e.g. .hin and .mol formats), has strong limitations in the number of SRUs that it polymerizes, not being sufficient for the polymer complexity (size and structure) existing in our databases [14]. On the other hand, Amsterdam Modeling Suite can import data in SMILES code (.smi) as inputs, but the output is in connection table format

[1)] Planta Piloto de Ingeniería Química (PLAPIQUI), Universidad Nacional del Sur (UNS) - Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Bahía Blanca, Buenos Aires, Argentina

[2)] Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC), Argentina

[3)] ISISTAN (CONICET – UNCPBA), Tandil, Buenos Aires, Argentina

[4)] Instituto de Ciencias e Ingeniería de la Computación (ICIC), (UNS - CONICET), Bahía Blanca, Buenos Aires, Argentina

[5)] Departamento de Ciencias e Ingeniería de la Computación, (DCIC-UNS), Bahía Blanca, Buenos Aires, Argentina

[6)] Departamento de Ingeniería Química (DIQ-UNS), Bahía Blanca, Buenos Aires, Argentina

[*)] Author for correspondence: mdiaz@plapiqui.edu.ar

(3D) and it requires long time to generate polymer chains with the required high degree of polymerization. This excessive execution time for calculating 3D macromolecules represents a drawback in the case of complex polymeric molecules with very high molecular weight. Finally, QuantumATK cannot deal with complex molecular structures as those present in our database.

In the last decade, we have been working with complex polymeric materials predicting mechanical properties derived from tensile test [15]. Research has evolved into increasingly realistic scenarios, and at this moment we are addressing the issue of polydispersity [10, 17]. The average molecular weights (Mw) of the polymeric molecules in our database are between 10^4 and 10^6 g/mol [15]. These materials have the characteristic of being polydisperse and also the different molecules that compose them are entangled with each other. For this reason, a 3D representation of a single molecular chain would not represent "the bulk material" and the computational effort associated with this representation would not be justified. For our QSPR predictive modeling objectives, 3D molecular descriptors (which are computed on 3D structures) are not necessary; moreover, they are not desirable because, when stabilizing the 3D structure, it adopts a shape that is very far from reality. In other words, it does not represent the entanglements of "the bulk material". Consequently, our investigations are based on 2D structures which, although they do not describe the three-dimensional reality, are able to capture important structural information that is independent of 3D.

Due to mentioned disadvantages and limitations of the proprietary software tools, we set out to develop an algorithm to build representations of high molar mass (MM) polymers based on simple molecular representation codes. The PolyMaS software tool aims to generate macromolecules, in short time, from SRU expressed in a computational representation. The output of this computation is a single 2D polymeric chain in SMILES code with the desired polymerization degree (PD). In the following section, a detailed explanation of the algorithm is provided.

SOFTWARE DESIGN

PolyMaS uses the Simplified Molecular Input Line Entry Specification (SMILES) [18] as a molecular representation format. This specification describes, in a simple and unambiguous way, the structure of a molecule. SMILES is an easy-to-interpret language for both informatics and chemists, because it uses ASCII characters to describe molecules.

Using the formal grammar [19] with which any SMILES is build, we can explain the generation of macromolecules process of PolyMaS. A SMILES string consists of a chain made up of atoms followed by a terminator. The symbol '*' is also accepted as a valid atom and represents a "wildcard". Some sentences that make up the grammar are included below in order to understand the PolyMaS algorithm (see Table 1).

The '*' represents an atom whose atomic number is unknown or unspecified. It does not have any specific electronic properties or valence. If specified outside square brackets, it takes on the valence implied by its bonds. If it is inside square brackets, it takes on the valence implied by its bonds, hydrogens, and/or charge. The full formal grammar can be seen in the OpenSMILES website [19].

The algorithm input is the SRU computational representation in SMILES format. Asterisks (*) are used as indicators that locate both the head and tail of the SRU. Once these indicators are located, a head-tail polymerization between two SRUs is imitated. The result is a computational representation of a single linear polymer chain in SMILES format.

As an example, Figure 1 shows the PolyMaS methodology for the generation of a single chain of polystyrene (SRU: C(C)c1ccccc1) with the desired polymerization degree (PD).

The SMILES chain complies with the grammar described above. The example for polystyrene includes the following grammar items:

– Two carbon atoms indicated with the uppercase letter C

– Branches indicated with parenthesis'()'

– An aromatic ring within the parenthesis indicated with 'c1ccccc1'

– The "wildcard" indicated with asterisks '*'.

PolyMaS uses the "wildcard" to identify the head and tail of the SRU (Figure 1; Step 1): *C(C*)c1ccccc1. It saves a copy without the first '*', that is, without the head (Figure 1; Step 2). It replaces the SRU tail (second '*') from step 1 by the SRU saved in step 2 (Figure 1; Step 3). Thus, a new SMILES chain consisting of two SRUs is obtained as follows: *C(CC(C*)c1ccccc1)c1ccccc1. This new chain also complies with the SMILES formal grammar. In this way, PolyMaS repeats the process until the desired PD is obtained (Figure 1; Steps 4-5).

RESULTS AND DISCUSSIONS

Software specifications and dependencies

The algorithm is implemented in the R language. It requires the *doParallel* and *foreach* libraries because the code is suitable for execution, taking advantage of parallel computing. The benefits of parallel computation are more evident when several molecules of large polymerization degrees are generated. However, the software is uploaded to a web server so that installation or dependencies are not necessary. In this sense, our software is operating system agnostic. PolyMaS is licensed by GNU GPL and can be used in a very simple way through this web page: http://lidecc.cs.uns.edu.ar/ChIT/ALIS/PolyMaS/

T a b l e 1. Formal grammar for SMILES code

Formal Grammar
atom ::= bracket_atom | aliphatic_organic | aromatic_organic | '*'
aliphatic_organic ::= 'B' | 'C' | 'N' | 'O' | 'S' | 'P' | 'F' | 'Cl' | 'Br' | 'I' **aromatic_organic** ::= 'b' | 'c' | 'n' | 'o' | 's' | 'p'
bracket_atom ::= '[' isotope? symbol chiral? hcount? charge? class? ']' **symbol** ::= element_symbols | aromatic_symbols | '*' **isotope** ::= NUMBER **element_symbols** ::= 'H' | 'He' | 'Li' | 'Be' | 'B' | 'C' | 'N' | 'O' | 'F' | 'Ne' | 'Na' | 'Mg' | 'Al' | 'Si' | 'P' | 'S' | 'Cl' | 'Ar' | 'K' | 'Ca' | 'Sc' | 'Ti' | 'V' | 'Cr' | 'Mn' | 'Fe' | 'Co' | 'Ni' | 'Cu' | 'Zn' | 'Ga' | 'Ge' | 'As' | 'Se' | 'Br' | 'Kr' | 'Rb' | 'Sr' | 'Y' | 'Zr' | 'Nb' | 'Mo' | 'Tc' | 'Ru' | 'Rh' | 'Pd' | 'Ag' | 'Cd' | 'In' | 'Sn' | 'Sb' | 'Te' | 'I' | 'Xe' | 'Cs' | 'Ba' | 'Hf' | 'Ta' | 'W' | 'Re' | 'Os' | 'Ir' | 'Pt' | 'Au' | 'Hg' | 'Tl' | 'Pb' | 'Bi' | 'Po' | 'At' | 'Rn' | 'Fr' | 'Ra' | 'Rf' | 'Db' | 'Sg' | 'Bh' | 'Hs' | 'Mt' | 'Ds' | 'Rg' | 'Cn' | 'Fl' | 'Lv' | 'La' | 'Ce' | 'Pr' | 'Nd' | 'Pm' | 'Sm' | 'Eu' | 'Gd' | 'Tb' | 'Dy' | 'Ho' | 'Er' | 'Tm' | 'Yb' | 'Lu' | 'Ac' | 'Th' | 'Pa' | 'U' | 'Np' | 'Pu' | 'Am' | 'Cm' | 'Bk' | 'Cf' | 'Es' | 'Fm' | 'Md' | 'No' | 'Lr' **aromatic_symbols** ::= 'b' | 'c' | 'n' | 'o' | 'p' | 's' | 'se' | 'as'
chiral ::= '@' | '@@' | '@TH1' | '@TH2' | '@AL1' | '@AL2' | '@SP1' | '@SP2' | '@SP3' | '@TB1' | '@TB2' | '@TB3' | … | '@TB20' | '@OH1' | '@OH2' | '@OH3' | … | '@OH30' | '@TB' DIGIT DIGIT | '@OH' DIGIT DIGIT
hcount ::= 'H' | 'H' DIGIT
charge ::= '-' | '-' DIGIT? DIGIT | '+' | '+' DIGIT? DIGIT | '--' **deprecated** | '++' **deprecated**
class ::= ':' NUMBER
bond ::= '-' | '=' | '#' | '$' | ':' | '/' | '\' **ringbond** ::= bond? DIGIT | bond? '%' DIGIT DIGIT **branched_atom** ::= atom ringbond* branch* **branch** ::= '(' chain ')' | '(' bond chain ')' | '(' dot chain ')' **chain** ::= branched_atom | chain branched_atom | chain bond branched_atom | chain dot branched_atom **dot** ::= '.'
smiles ::= terminator | chain terminator **terminator** ::= SPACE | TAB | LINEFEED | CARRIAGE_RETURN | END_OF_STRING

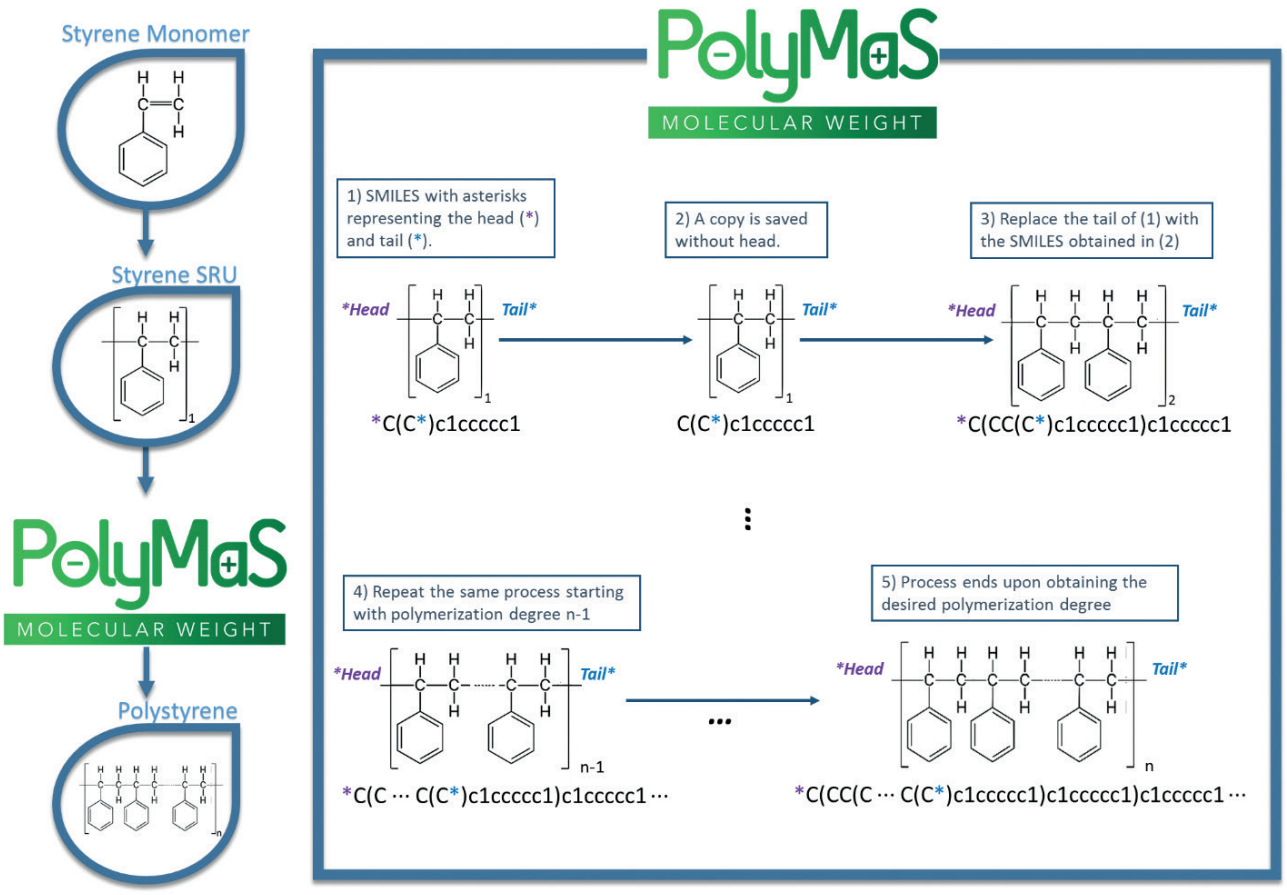

Fig 1. PolyMaS algorithm methodology using the polystyrene SRU as an example. In step 1, the head and tail are identified: *C(C*) c1ccccc1. In step 2, a copy is saved without the first '*', that is, without the head. In step 3, the SRU tail of step 1 is replaced by the SRU saved in step 2. Therefore, a new SMILES chain consisting of two SRUs is obtained as follows: *C(CC(C*)c1ccccc1)c1ccccc1. In steps 4 and 5, this process is repeated until the desired polymerization degree (PD) is obtained

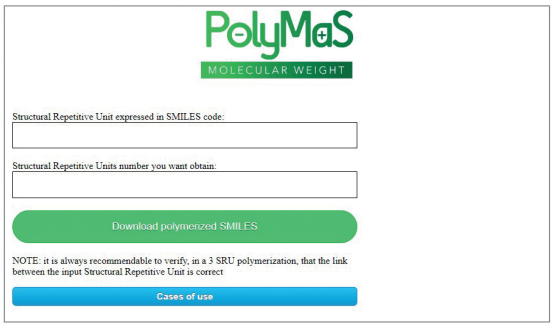

Fig 2. PolyMaS testing web form (http://lidecc.cs.uns.edu.ar/ChIT/ALIS/PolyMaS/PolyMaS.php).

PolyMaS.php. The internet address for downloading the package for R language and running it in a local way is: https://github.com/AlchimiaInSilico/PolyMaS (installation instructions are included).

Macromolecule generation example

PolyMaS has two input parameters: SRU and PD. An instruction of use is available at: http://lidecc.cs.uns.edu.ar/ChIT/ALIS/PolyMaS/PolyMaS-case-of-use.php. A web form with two fields is presented there (Figure 2). In the first field, enter the SRU, that is, the SMILES string with two asterisks (*) used as indicators of the head (the first one) and the tail (the second one). The polystyrene SRU in SMILES code is provided as an example: *C(C*)c1ccccc1. In the second field, enter the desired PD, expressing the number of SRUs of the molecule to be generated. Although PolyMaS does not have length limits, the web server where it is hosted has an operating time limit; for this reason, there could be a timeout error for too high PDs. Nevertheless, this limitation should not occur when PolyMaS is executed on a local computer.

Finally, click on "Download polymerized SMILES" and a file called "PolyMaS_smiles.smi" containing the resulting macromolecule expressed by their computational representation (SMILES code) will be downloaded. This file can be used directly for a variety of descriptor calculation tools or any application that supports the SMILES format.

Table 2 shows some polymers in our database [15], their average molecular weights (Mw), and the number of SRUs (PD) needed to reach a weight (M) nearly Mw. In addition, it shows the execution time (elapsed) required by PolyMaS to generate, for each polymer, the macromolecule whose obtained molecular weight (M) tries to meet the order of magnitude of the corresponding Mw of each example. The execution was under the following hardware specifications: 8GB RAM and Intel CPU (R) i3-3110 2.40 GHz, reporting an average runtime of 0.182 [s] with a standard deviation of 0.141 [s].

CONCLUSIONS

In this paper, we present PolyMaS, a software tool that generates macromolecules starting from a SRU, with

T a b l e 2. Example of PolyMaS' testing for five polymers. It shows the SRU's IDs and their molar mass (MM) from our database [15], polymer's names, SMILES codes, Mw, numbers of SRUs (PDs) needed to meet the Mw magnitude order, the obtained molecular weight (M), and the corresponding execution times (Elapsed).

ID	MM g/mol	Polymer	SMILES code	M_w g/mol	PD	M* g/mol	Elapsed s
3	104.16	Polystyrene	*C(C*)c1ccccc1	880 000	8448	879 982	0.36
24	170.14	Poly([2-(trifluoromethyl)phenyl]acetylene)	*C(=C*)c1c(cccc1)C(F)(F)F	690 000	4055	689 937	0.31
38	634.66	Poly([4,4'-(9H-fluorene-9,9-diyl)dianiline]-alt-[5,5'-carbonylbis(isobenzofuran-1,3-dione)])	*N1C(=O)c2c(C1=O)cc(c2)C(=O)c1ccc2c(=O)N(c(=O)c2c1)c1ccc(cc1)C1(c2ccccc2c2ccccc12)c1ccc(cc1)*	26 500	42	26 658	0.08
56	1103.34	Poly({4,4'-[tricyclo[5.2.1.0^2,6^^]decane-8,8-diyl-bis(4,1-phenyleneoxy)]dianiline}-alt-[5,5'-[4-phenylcyclohexane-1,1-diyl]bis(4,1-phenyleneoxy)]diiso-benzofuran-1,3-dione])	*c1ccc(cc1)Oc1ccc(cc1)[C@@]1(C[C@H]2C[C@@H]1[C@H]1[C@@H]2CCC1)c1ccc(cc1)Oc1ccc(cc1)N1C(=O)c2c(C1=O)cc(cc2)Oc1ccc(cc1)[C@@]1(CC[C@H](CC1)c1ccccc1)c1cc-c(cc1)Oc1ccc2c(c1)C(=O)N(C2=O)*	94 000	85	93 786	0.07
77	624.82	Poly[sulfonyl(3-sulfo-1,4-phenylene)sulfanediyl-1,4--phenylenesulfanediyl-1,4-phenylenesulfanediyl(2--sulfo-1,4-phenylene)]	*S(=O)(=O)c1cc(S(=O)(=O)c(cc1)Sc1ccc(cc1)Sc1ccc(cc1)Sc1c(S(=O)(=O)O)cc(cc1)*	158 000	253	158 083	0.09

* hydrogen-end-capped.

no size limit. For this task, only the SRUs expressed in SMILES format and the desired polymerization degree must be known. By entering these two input parameters, the software generates the macromolecule with the chain expressed in SMILES code.

Using PolyMaS, the possibility of obtaining a computational representation of high molar mass chains becomes feasible. In addition, most of molecular descriptor calculation tools support the PolyMaS output. This feature allows the calculation of descriptors for a large variety of high molar mass chains, which leads to more realistic databases for linear polymers with polydispersity.

The software is available on the web; and also, if the user prefers, it can be installed locally. To the best of our knowledge, PolyMaS is the fastest tool available today to obtain the desired polymerization degree for the 2D representation of high molar mass polymers. Furthermore, since the tool is freely available as an R package, it offers integration possibilities in a work pipeline, speeding up and automatizing the task for large databases. This last point represents an additional advantage with respect to the functionalities offered by other tools.

ACKNOWLEDGMENTS

This work was partially supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina, [Grant N° PIP 112-2017-0100829], the Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina, [Grants N° PGI 24/N042 and PGI 24/ZM17] and the Agencia Nacional de Promoción Científica y Tecnológica [Grants PICT 2018-04533 and PICT-2019-03350]. We thank Rudy Garzotto for designing the PolyMaS logo, and finally Dr. Claudio Javier Pérez and Dr. Miriam Cristina Strumia for testing PolyMaS.

REFERENCES

[1] Kim C., Chandrasekaran A., Huan T.D. *et al*.: *The Journal of Physical Chemistry C* **2018**, *122*(*31*), 17575. https://doi.org/10.1021/acs.jpcc.8b02913

[2] Toropov A.A., Toropova A.P., Kudyshkin V.O. *et al*.: *Structural Chemistry* **2020**, *31*(*5*), 1739. https://doi.org/10.1007/s11224-020-01588-8

[3] Sha W., Li Y., Tang S. *et al*.: *InfoMat* **2021**, *3*(*4*), 353. https://doi.org/10.1002/inf2.12167

[4] Chen L., Pilania G., Batra R. *et al*.: *Materials Science and Engineering: R: Reports* **2021**, *144*, 100595. https://doi.org/10.1016/j.mser.2020.100595

[5] https://www.polymergenome.org (access date 17.03.2021).

[6] Cherkasov A., Muratov E.N., Fourches D. *et al*.: *Journal of Medicinal Chemistry* **2014**, *57*(*12*), 4977. https://doi.org/10.1021/jm4004285

[7] Adams N.: "Polymer informatics. In: Meier M., Webster D. (eds) Polymer Libraries", Springer, Berlin, Heidelberg 2010, pp. 107–149. https://doi.org/10.1007/12_2009_18

[8] Cravero F., Martínez M.J., Ponzoni I., Díaz M F.: *Chemometrics and Intelligent Laboratory Systems* **2019**, *193*, 103851. https://doi.org/10.1016/j.chemolab.2019.103851

[9] Himanen L., Geurts A., Foster A. S., Rinke P.: *Advanced Science* **2019**, *6(21),* 1900808. https://doi.org/10.1002/advs.201900808

[10] Cravero F., Schustik S.A., Martínez M.J. *et al*.: *Chemometrics and Intelligent Laboratory Systems* **2019**, *191*, 65. https://doi.org/10.1016/j.chemolab.2019.06.006

[11] https://www.hyper.com (access date 17.03.2021).

[12] https://www.scm.com (access date 17.03.2021).

[13] https://www.synopsys.com/silicon/quantumatk.html (access date 17.03.2021).

[14] Palomba D., Vazquez G.E., Díaz M.F.: *Journal of Molecular Graphics and Modelling* **2012**, *38*, 137. https://doi.org/10.1016/j.jmgm.2012.04.006

[15] Palomba D., Vazquez G.E., Díaz M.F.: *Chemometrics and Intelligent Laboratory Systems* **2014**, *139*, 121. https://doi.org/10.1016/j.chemolab.2014.09.009

[16] Cravero F., Martínez M.J., Vázquez G.E. *et al*.: *Journal of Integrative Bioinformatics* **2016**, *13*(*2*), 286. https://doi.org/10.1515/jib-2016-286

[17] Cravero F., Schustik S.A., Martínez M.J. *et al*.: *Journal of Chemical Information and Modeling* **2020**, *60(2),* 592. https://doi.org/10.1021/acs.jcim.9b00867

[18] Weininger D.: *Journal of Chemical Information and Computer Sciences* **1988**, *28(1),* 31. https://doi.org/10.1021/ci00057a005

[19] http://opensmiles.org (access date 17.03.2021).

Received 23 III 2021.