SoftwareX 15 (2021) 100749

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Software update

Update 1.1 to "pysimm: A python package for simulation of molecular systems", (PII: S2352711016300395)

Alexander G. Demidov [a,b,c], B. Lakshitha A. Perera [a,b,c], Michael E. Fortunato [a,b,c], Sibo Lin [d], Coray M. Colina [a,b,c,e,*]

[a] Department of Chemistry, University of Florida, Gainesville, FL 32611, United States
[b] George and Josephine Butler Polymer Research Laboratory, University of Florida, Gainesville, FL 32611, United States
[c] Center for Macromolecular Science & Engineering, University of Florida, Gainesville, FL 32611, United States
[d] Aramco Services Company: Aramco Research Center — Boston, 400 Technology Square, Cambridge, MA, 02139, United States
[e] Department of Materials Science and Engineering, University of Florida, Gainesville, FL 32611, United States

ARTICLE INFO

Article history:
Received 21 June 2021
Accepted 23 June 2021

Keywords:
Molecular simulations
Polymer models
Pysimm

ABSTRACT

This update of the pysimm application programming interface, pysimm 1.1, provides both infrastructural as well as functional updates. Moreover, improvements to the random walk application that allow it to construct polymers with controlled tacticity are highlighted. Additions to the forcefield module include an update to enable working with the family of CHARMM forcefields and automated typing with the CHARMM generalized forcefield (CGenFF). Finally, new detailed examples demonstrating new features are also provided.

Code metadata

Current code version	*v1.1*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-21-00117
Code Ocean compute capsule	
Legal Code License	*MIT*
Code versioning system used	*git*
Software code languages, tools, and services used	*Python 3.x*
Compilation requirements, operating environments & dependencies	*Linux*
If available Link to developer documentation/manual	https://pysimm.readthedocs.io
Support email for questions	support@pysimm.org

1. Motivation and significance

Pysimm is an open-source object-oriented Python package for molecular simulations [1]. It handles data organization for particles, force field parameters, and simulation settings so that the user can focus on developing their simulation workflow. The version 0.2 release was mainly focused on providing an interface to CASSANDRA (a Monte Carlo molecular simulation software), that allowed the possibility to perform hybrid simulations through the connection with the LAMMPS simulation package for Molecular Dynamics. The focus point of the current release, pysimm 1.1, is to summarize new improvements that include both infrastructure (migrated to Python 3.x, a pysimm Docker image was introduced, and integration tests) as well as functionality update (additions to the forcefield module, expansion of the random walk application, and an interface to the Zeo++ software package [2]). The forcefield module was updated to enable work with the family of CHARMM forcefields [3–5]. Finally, application tools were added to control and analyze the tacticity of polymer chains.

The swift growth of experimental [6] and computational [7] research in the area of polymer-protein bioconjugates in the past decades has emphasized the need for sophisticated computational models to extract essential molecular level information and to inform experimental efforts [8]. Some computational tools, such as the CHARMM-GUI [9], can facilitate the construction of polymers for bioconjugate systems, however, only a limited number of polymers are available in their libraries, and the models are oftentimes based on one specific forcefield. The expanded

DOI of original article: https://doi.org/10.1016/j.softx.2016.12.002.
* Corresponding author at: Department of Chemistry, University of Florida, Gainesville, FL 32611, United States.
E-mail address: colina@chem.ufl.edu (Coray M. Colina).

https://doi.org/10.1016/j.softx.2021.100749
2352-7110/

functionality of pysimm further extends the variety of polymers available for the computational bioconjugates models by allowing the users to construct any polymer of interest and choose the most appropriate (among the available in pysimm) forcefield for the task. The stereochemistry of adjacent monomer units, or tacticity, has notable effects on polymer material properties [10]. The updated functionality of pysimm allows for tabulation and generation of such polymers with defined tacticity. In particular, this functionality has been recently applied to the simulation of vinyl-type polynorbornene models that possess cis-disyndiotactic backbones [11], a complicated stereochemistry that was ignored in prior modeling studies.

2. General code improvements

The pysimm code has been migrated to Python 3.x. A series of integration and unit tests compatible with the PyTest framework were added and should be useful for further pysimm development. The pysimm distribution now contains a Dockerfile to create and deploy a Docker image containing the latest pysimm version integrated with the molecular simulation software it can interface to, i.e., LAMMPS [12], CASSANDRA [13], PoreBlazer [14], PyIAST [15] and Zeo++ [2]. The 'pysimm.zeopp' application translates a pysimm system into the format acceptable by Zeo++ and generates the remaining input files for the various analysis modes. The updated pysimm interface now can work with the latest Poreblazer versions (3.0.x, and 4.0) as well as the earlier versions (2.x) [14].

3. Updates to the forcefield module: simulations with the CHARMM Ether and CGenFF forcefields

An advantage of the 'pysimm.forcefield' module is the functionality of an automatic condition-based atom typing procedure, that is available for the Dreiding [16], GAFF [17], PCFF [18], and now the CHARMM general force field (CGenFF). This implies that parameters of all bonded and nonbonded interactions will be assigned automatically and the recognition of some common atom types described in CGenFF (for carbons, hydrogens, oxygens, nitrogens, and sulfurs) are now available. The closest chemical surrounding of an atom is analyzed to derive the correct atom type and the presence of explicit van der Waals parameters for nonbonded interactions between different atom types available in the CHARMM parameters database [19], are accounted for in the 'forcefield.Charmm' class.

The parameters of each forcefield available in pysimm are kept in a database file, and the format of the parameters adopted is the same as in the LAMMPS simulation package. Thus, parameters of all terms of the CHARMM forcefield, except the dihedrals, are presented in this format. Special attention should be taken when transforming parameters of dihedral terms within the force field functional forms implemented in different packages. For example, some dihedrals are described in CHARMM by a sum of several cosine terms whereas standard LAMMPS representation of CHARMM dihedrals is a single cosine term. Thus, the CHARMM dihedrals in pysimm are presented as more general Fourier dihedrals. Along with CGenFF, the parameters adopted by pysimm include CHARMM ether, CHARMM all-hydrogen for proteins, CHARMM carbohydrate, and CHARMM lipid parameters. An illustrative example of MD simulations of a single polyethylene oxide (PEO) chain in water typed with two different sets of CHARMM forcefield parameters (CGenFF and CHARMM ether) is discussed later in this work.

Along with PEO, the repetitive units of a few other polymers are added to the monomer library of pysimm: pQA [poly (quaternary ammonium methacrylate)], pSMA [poly(sulfonate methacrylate)], poly(carboxybetaine methacrylate) [pCBMA], and polynorbornene [pNB] (for ring-opening metathesis polymerization). These repeating units are of interest to the bioconjugate community [20,21] and are provided as examples of the construction of polymer chains with bulkier side chains, or charged side chains using pysimm random walk. Even though the majority of bonded parameters are provided by the CHARMM database, and thus are known to pysimm, there might be instances in which a system has a bonded parameter (e.g., dihedral) not listed in the database. In that case, the user can modify the pysimm forcefield data similarly to the data in any text file: either by copying and updating an existing record or creating a new one.

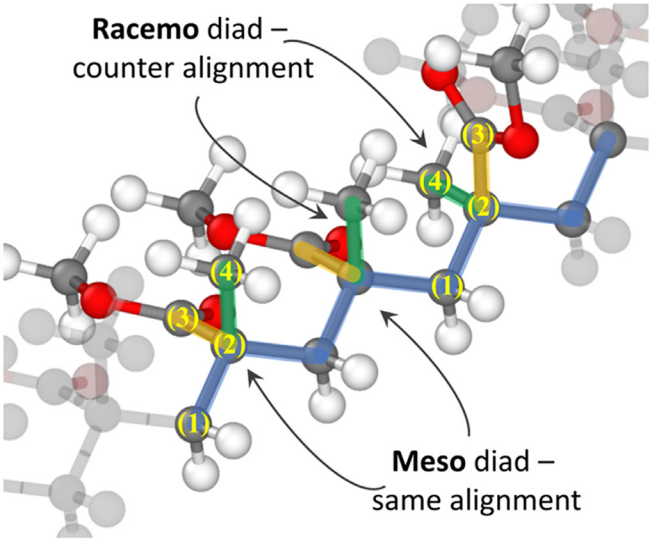

Fig. 1. An Illustration of a diad-based tacticity definition in a polymer chain utilized in the 'check_tacticity()' function of 'pysimm.random_walk'.

4. Updates for the random walk application

Previously the random walk application exposed two functions 'random_walk()' and 'copolymer()' which had similar signatures and were able to construct head-to-tail polymer chains from a single monomer, or a set of monomers, respectively. This update adds another function: 'check_tacticity()' which analyzes the tacticity of a single polymer chain. The algorithm uses the diad-based definition of tacticity and outputs a sequence of Booleans that label whether the next diad along the backbone is meso (1) type or racemo (0) type (see Fig. 1, for which the function will output sequence of [1, 0]).

A second function 'random_walk_tacticity()' allows the user to define the proportion of meso diads to create a macromolecule via the 'tacticity' keyword setting. The function will return a polymer chain, created "in vacuum" with the predetermined tacticity. However, it is important to note that this tacticity might not be preserved after the polymer chain is passed through a minimization and equilibration processes. If a predetermined tacticity is desired, we suggest the following workflow: (1) run 'random_walk_tacticity' without forcefield optimization that will concatenate repetitive units and put them in approximately correct positions, and (2) run a long molecular dynamics simulation for the system to relax the atoms' positions. Finally, for the 'random_walk()' and 'copolymer()' functions it is now possible to define a different configuration (rotation, translation, or reflection) of the next repetitive unit to be added to the chain during the random walk procedure. Both functions accept the 'extra_bonds' parameter which enables the growth of a polymer

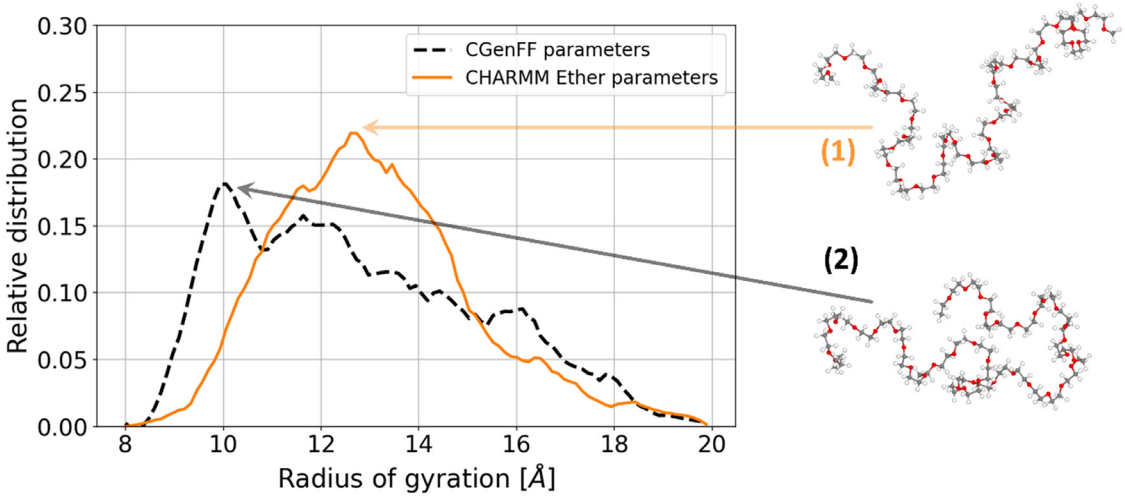

Fig. 2. The distributions of the radius of gyration of a single PEO chain in water. Two 50 ns molecular dynamics simulations were set up with pysimm and were run in LAMMPS using 2 sets of CHARMM parameters, CGenFF and CHARMM ether. The polymer chains on the side show one of the most abundant configurations for each parameter set.

chain with repetitive units connected by more than one chemical bond.

5. Example

The general workflow of a MD simulation setup with the CHARMM forcefield is illustrated in example #13 of the distribution. In this example, a short (37 repetitive units) PEO chain is created, and then solvated in cTIP3P water [22–24]. Pysimm then sets up the MD simulations to equilibrate the whole system and run production simulations. In this example, the chain is typed with either the CGenFF parameters or the CHARMM-ether forcefield parameters [25]. The example showcases the improved implementation of the pysimm forcefield typing scheme, which now allows the users to explicitly set types for certain particles in the system by labeling those particles with the corresponding type names.

Two similar PEO chains, typed with two different sets of parameters (CHARMM Ether or CGenFF), were subjected to two otherwise identical 50 ns MD simulations in LAMMPS, controlled by pysimm. To evaluate the effect of the selected parameter set on the final results, the distributions of the Radius of Gyration (R_g) of the two chains were obtained. R_g is a parameter often utilized to determine a polymer chain's compactness. Interestingly, a noticeable difference was observed between the R_g distributions calculated for the two chains (Fig. 2). Previously it has been reported that the average R_g of a PEO chain, in similar conditions, is equal to 13.0 $\pm$ 2.4 Å [26]. While both simulations show average R_g values close to the reference, there are differences in the R_g distributions between the two cases. While the chain modeled with the CHARMM Ether FF showed a unimodal distribution with a peak approximately at the average value, the chain modeled with the CGenFF show a skewed distribution with most abundant R_g configuration at 10 Å and a prolonged tail from 12 to 18 Å. It is worth noting that while the CHARMM Ether and CGenFF forcefield parameters are very similar for PEO, even the small differences in the parameters can result in a significant difference in certain thermodynamic properties [27,28].

6. Conclusions

Pysimm is a user-friendly, versatile, and up-to-date Python API which facilitates the design of molecular simulations on different research fields, from nanoporous materials to protein-polymer bioconjugates. The latest update featured in this work follows that direction, and provides modifications of the random walk application which gives the user more control over the local geometry of repetitive units and expand the abilities of pysimm to make polymer models (single chains in solvent or multi-chain amorphous samples, blends, in solution, or below their glass transition temperature). The update is also aimed to help with the construction of polymer-protein bioconjugate models. The examples presented in this work have been added to the pysimm code repository (https://github.com/polysimtools/pysimm) and set useful templates for the construction of polymer chains with desired tacticity distributions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to acknowledge the U.S. Department of Energy for funding (grant DE-FG02-17ER16362).

References

[1] Fortunato ME, Colina CM. Pysimm: A python package for simulation of molecular systems. SoftwareX 2017;6:7–12.
[2] Willems TF, Rycroft CH, Kazi M, Meza JC, Haranczyk M. Algorithms and tools for high-throughput geometry-based analysis of crystalline porous materials. Microporous Mesoporous Mater 2012;149(1):134–41.
[3] Mackerell AD, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. J Phys Chem B 1998;(102):3586–616.
[4] Vanommeslaeghe K, et al. CHARMM General force field (cgenff): A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem 2010;31(4):671–90.
[5] Vanommeslaeghe K, Raman EP, Mackerell JDA. Automation of the CHARMM general force field (cgenff) II: Assignment of bonded parameters and partial atomic charges. J Chem Inf Model 2012;(52):3155–68.
[6] Kaupbayeva B, Russell AJ. Progress in polymer science polymer-enhanced biomacromolecules. Prog Polym Sci 2020;101:101194.
[7] Munasinghe A, Baker SL, Lin P, Colina CM, Russell AJ. Structure–function–dynamics of a-chymotrypsin based conjugates as a function of polymer charge. Soft Matter 2020;16:456–65.
[8] Russell AJ, et al. Next generation protein-polymer conjugates. AIChE J 2018;64(9):3230–45.
[9] Choi YK, et al. CHARMM-GUI Polymer builder for modeling and simulation of synthetic polymers. J Chem Theory Comput 2021;17:2431–43.

[10] Coates GW. Precise control of polyolefin stereochemistry using single-site metal catalysts. Chem Rev 2000;(100):1223–52.
[11] Sundell BJ, et al. Exo-selective, reductive heck derived polynorbornenes with enhanced molecular weights, yields, and hydrocarbon gas transport properties. ACS Macro Lett 2020;9(9):1363–8.
[12] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. J Comput Phys 1995;117(1):1–19.
[13] Shah JK, et al. Cassandra: An open source Monte Carlo package for molecular simulation. J Comput Chem 2017;1727–39.
[14] Sarkisov L, Bueno-perez R, Sutharson M, Fairen-jimenez D. Materials informatics with poreblazer v4.0 and the CSD mof database. Chem Mater 2020;32:9849–67.
[15] Simon CM, Smit B, Haranczyk M. Pyiast: Ideal adsorbed solution theory (IAST) python package. Comput Phys Comm 2016;200:364–80.
[16] Mayo SL, Olafson BD, Goddard WA. DREIDING: A generic force field for molecular simulations. J Phys Chem 1990;94(26):8897–909.
[17] Wang J, Wolf R, Caldwell J, Kollman P, Case D. Development and testing of a general AMBER force field. J Comput Chem 2004;34:1157.
[18] Sun H. Force field for computation of conformational energies, structures, and vibrational frequencies of aromatic polyesters. J Comput Chem 1994;15(7):752–68.
[19] Mackerell lab. 2021, [Online]. Available: http://mackerell.umaryland.edu/charmm_ff.shtml [Accessed: 25-May-2021].
[20] Baker SL, et al. Intramolecular interactions of conjugated polymers mimic molecular chaperones to stabilize protein - polymer conjugates. Biomacromolecules 2018;19:3798–813.
[21] Lin P, Colina CM. Molecular simulation of protein–polymer conjugates. Curr Opin Chem Eng 2019;23:44–50.
[22] Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. Comparison of simple potential functions for simulating liquid water liquid water. J Chem Phys 1983;79(2):926–35.
[23] Mark P, Nilsson L. Structure and dynamics of the TIP3P, SPC, and SPC/E water models at 298 K. J Phys Chem A 2001;(105):9954–60.
[24] Price DJ, Brooks CL. A modified TIP3P water potential for simulation with ewald summation. J Chem Phys 2004;121:10096.
[25] Baker CM, MacKerell J, D A. Polarizability rescaling and atom-based thole scaling in the CHARMM drude polarizable force field for ethers. J Mol Model 2010;16(3):567–76.
[26] Rukmani SJ, Kupgan G, Anstine DM, Colina CM. A molecular dynamics study of water-soluble polymers: analysis of force fields from atomistic simulations. Mol Simul 2019;7022:310–21.
[27] Zhu X, Lopes PEM, Mackerell Jr AD. Recent developments and applications of the CHARMM force fields. WIREs Comput Mol Sci 2012;2(1):167–85.
[28] Chowdhary J, Harder E, Lopes PEM, Huang L, Mackerell Jr AD. A polarizable force field of dipalmitoylphosphatidylcholine based on the classical drude model for molecular dynamics simulations of lipids. J Phys Chem B 2013;117:9142–60.