Efficient Computational Chemistry Protocol Implementations for Chemical Research in Pandemic Situation

Aditya Wibawa Sakti[1], Nila tanyela[2]

{aditya.ws@universitaspertamina.ac.id[1], nila.tanyela@universitaspertamina.ac.id[2]}

Universitas Pertamina Jl. Teuku Nyak Arief, RT.7/RW.8, Simprug, Kec. Kby. Lama, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12220, Indonesia[1,2]

Abstract. During the COVID-19 pandemic situation, accessing a chemical laboratory becomes problematic. The issues emerge not only for experimentalists but also for computational chemists who require additional access to the computer hardware for maintenance purposes. As a result, a predictive computational chemistry study is required before conducting laboratory research, which necessitates access to the laboratory. Therefore, in the present work, an efficient and user-friendly command-line interface software is developed to speed up the whole research in the field of Computational Chemistry. The software can be installed easily in elastic computing systems, personal computers, and other high-performance computing environments. The software employed both classical and quantum mechanics to perform fast calculations that could quickly predict physical and chemical properties. Therefore, by using the implemented tools, the experimentalists can predict possible reaction routes quickly, reducing the time in the lab, and thus preventing the virus exposal.

Keywords: Computational chemistry, quantum mechanics, classical mechanics, density-functional tight-binding, molecular dynamics.

1 Introduction

Pandemic situation is defined as an epidemic condition that is occurring worldwide and thus leading to global infections [1] [2] [3]. The current pandemic situation also leads to a multidimensional challenge, including the challenge in performing scientific research, especially those that require *in-vitro* experiments [4] [5] and educations [6]. Accessing Chemical laboratories has been limited in several countries due to the lockdown. To overcome these issues, unnecessary protocols in the laboratory activities should be omitted and reduced for the sake of research efficiency. The main idea was previously implemented by developing online remote laboratories and various online-based studies [7] [8] [9] [10] [11]. In the Chemistry field, the major problem comes from how frequently one accesses the Chemical laboratory and performing *in-vitro* experiments.

ICONIC-RS 2022, March 31-April 01, DKI Jakarta, Indonesia

DOI 10.4108/eai.31-3-2022.2320672

Such an issue can be solved by performing computational study prior to the *in-vitro* experiments [12] [13]. Structural, energetic, and dynamical properties that are three pillars in Physical Chemistry can be calculated easily by using the recently developed free quantum chemical calculation packages, such as Orca [1] [15], GAMESS [16], DFTB+ [17], Dcdftbmd [18], and xTB [19] [20]. Similarly, free molecular mechanics-based software has also thoroughly developed, such as GROMACS [19-27], NAMD, and LAMMPS. Despite many open-source packages, managing input files to perform quantum chemical and classical calculations can be a high barrier for students or researchers who are not familiar with the software. To minimize the difficulty in making routine input files to perform similar computational jobs, a general code protocol has been developed.

The implemented code, namely, simulation protocol environment (SPE) has been developed to easily perform simulations and computations ranging from classical to quantum chemical levels. At present, the SPE code supports an interface for GROMACS and Dcdftbmd codes. Generally, the developed code is inspired by the atomistic simulation environment (ASE) code [28], yet with the function to parse minimal commands in the command line interface. In present work, the general structure of SPE will be introduced as well as two example cases in performing molecular dynamics simulations for sodium-ion battery design and predicting the reaction spontaneity.

2 Methodology

2.1 Structure preparation for the electrolyte of battery-ion system

Preparing a good initial structure for the solution system, such as the electrolyte of battery-ion system is of importance, especially to ensure the randomness of the initial structure for further molecular dynamics simulation in a higher level of theory. In the SPE code implementation, the initial structure of electrolyte solution is prepared from two main components, namely, solutes and solvent. For the case of battery-ion electrolyte, the solutes only consist of cation and anionic species, namely, Li^+ and trifluoromethanesulfonate (OTf^-) anion. On the other hand, 10 organic solvents that are commonly used for lithium-ion battery (LIB) electrolyte were adopted. All electrolyte models were prepared as such that the concentration of Li^+ in the solution is 1 mol/kg, namely, the concentration of the dilute electrolyte model. The user can also change the concentration of the electrolyte to create a model of superconcentrated electrolyte solution as appear in Ref. 29-32.

Initially, the electrolyte components were mixed in a cubic unit cell with a certain volume by using the modified PACKMOL program. The original program can be obtained in the previous work [33] [34]. The volume of unit cells depends on the molal concentrations of the adopted electrolyte solution. A series of molecular dynamics simulations at classical level is performed in a consecutive way, namely, beginning with energy minimization and followed by equilibrations under the microcanonical (*NVE*), canonical (*NVT*), and isotherm-isobaric (*NPT*) ensembles. Before performing the above-mentioned simulations, the force-field was parameterized by employing the generalized amber force-field (GAFF) format. The GAFF has been thoroughly used for describing thermodynamic and transport properties of various solutions [35]. After performing the *NPT* simulation, the mass density of the adopted system can be theoretically determined. A good initial structure can play a crucial role in further performing MD simulations by using the supercomputer or personal computer. The above protocol was performed by using the SPE code by simply specifying the type of the job, so that the user does not require a prior knowledge of molecular dynamics simulations nor the structure of the GROMACS input file.

Coordination numbers and diffusion coefficients can be calculated at classical mechanics level, namely, directly from the output files of the SPE program. The output files are generally standard GROMACS outputs that consist of trajectories, velocities, and other topological information. Diffusion coefficients were estimated based on the following Einstein equation:

$$D = \lim_{t\to\infty} \frac{\langle |r(t) - r(0)|^2 \rangle}{6t} \qquad (1)$$

In Eq. (1), $r(t)$and $r(0)$is atomic vector position at time t and time $t = 0$, respectively. To reproduce the diffusion coefficients correctly, however, will require more than one independent molecular dynamics (MD) trajectories. In the present work, as a case example, the diffusion coefficients from two independent trajectories.

2.2 Quantum chemical calculations for free Gibbs energy calculations

Gibbs free energy calculations were performed by a prior geometry optimization and Hessian matrix calculations. Both calculations were performed by using the Dcdftbmd integrated in the SPE code. Presently, the SPE code performance is evaluated for predicting the combustion reaction of C_nH_{2n+2} ($n = 1 - 100$) at high temperature to ensure that all products and reactants are in the gas phases. The computational time to compute the free Gibbs energy of combustion has been assessed. The Gibbs free energy of reaction was calculated for the following reaction:

$C_nH_{2n+2} + (3n+1)/2\ O_2 \rightarrow n\ CO_2 + (n+1)\ H_2O$

All calculations were performed at quantum mechanical level, namely, by employing the density-functional tight-binding (DFTB) Hamiltonian [36] [37] with the DFT-D4 [38] [39] dispersion correction. The Dcdftbmd code was used for optimizing and calculating Hessian matrices of each molecule involved in the combustion. The Hessian matrices were constructed to further calculate the thermal corrections as such that all Gibbs free energy were calculated at higher temperature. In this case, the room temperature of 298.15 K was used for calculating the vibrational frequencies, which then allowed us to calculate the thermal corrections.

3 Results and discussion

3.1 Semi-automatic electrolyte generator for lithium-ion battery application

The SPE code automatically adjusts the number of solvents and ions in the system simply by a command line interface. For this purpose, the user requires to input the type of solvent and counter-cation that is used in the present work. Note that the lithium ion was added automatically by the antechamber, parmchk2, and tleap combinations [40] [41]. Diffusion coefficients of lithium ion, TFSA ion, and the adopted organic solvents were estimated semiautomatically with the SPE code with the results shown in Table 1. The diffusion coefficient for a more concentrated solution is lower than that of the diluted counterpart. The estimated values agree with previous theoretical work that investigates the ion and solvent diffusivity by varying the salt concentration [29]. In general, in a dilute solution, the ion diffusions are always lower than that of the solvent diffusion, except for the case of VC. As shown in Table 1, increasing the concentration of VC from 1 mol/kg to 5 mol/kg also changes the tendency of a faster ionic diffusion. An ion-pair formation effect might be the reason behind this phenomenon [29].

Table 1. Calculated diffusion coefficients of lithium ion, TFSA anion, and the adopted organic solvents in 5 mol/kg electrolyte solutions. The previously estimated values [42] were also shown for 1 mol/kg of solutions. All diffusion coefficients are in 10^{-5} cm^2/s units.

Model	Organic solvent	D_{Li}		D_{TFSA}		D_{Solvent}	
		1 mol/kg	5 mol/kg	1 mol/kg	5 mol/kg	1 mol/kg	5 mol/kg
#1	Adiponitrile	0.325	0.029	0.359	0.032	0.647	0.088
#2	Acetonitrile	1.442	0.036	1.653	0.051	4.871	1.120
#3	Butylene carbonate	0.242	0.012	0.217	0.022	0.339	0.045
#4	Dimethoxyethane	0.468	0.026	0.631	0.026	1.811	0.102
#5	Dimethylsulfoxide	0.339	0.033	0.814	0.031	0.816	0.036
#6	Ethylene carbonate	0.277	0.020	0.224	0.024	0.604	0.052
#7	Ethylmethyl carbonate	0.394	0.017	0.727	0.017	0.851	0.047
#8	Propylene carbonate	0.234	0.011	0.483	0.018	0.477	0.047
#9	Tetrahydrofuran	0.548	0.039	1.152	0.035	2.642	0.088
#10	Vinylene carbonate	0.752	0.017	0.690	0.018	0.441	0.090

The computational cost to perform the above simulations was not so demanding. To obtain the diffusion coefficients as well as all mean-square-displacement (MSD) curves, it only took at most 7 minutes. The overall wall CPU time in seconds unit is shown in **Figure 1**. The wall CPU time is dominated by a long production run simulation. A smaller wall CPU time indicates a smaller system size. As an example, model #2 exhibits the smallest wall CPU time, which indicates that model #2 is the smallest electrolyte model in the present work. It is because the ATN molecule is the smallest among the other adopted solvents, thus leading to the fastest simulations. The simulations were performed in 8 processors laptop with the random-access memory of 8 GB. The simulations can be faster in the high-performance computing system. The present semi-automatic protocol helps to automate the same simulations to be consistent and free from a human error during input file preparations.

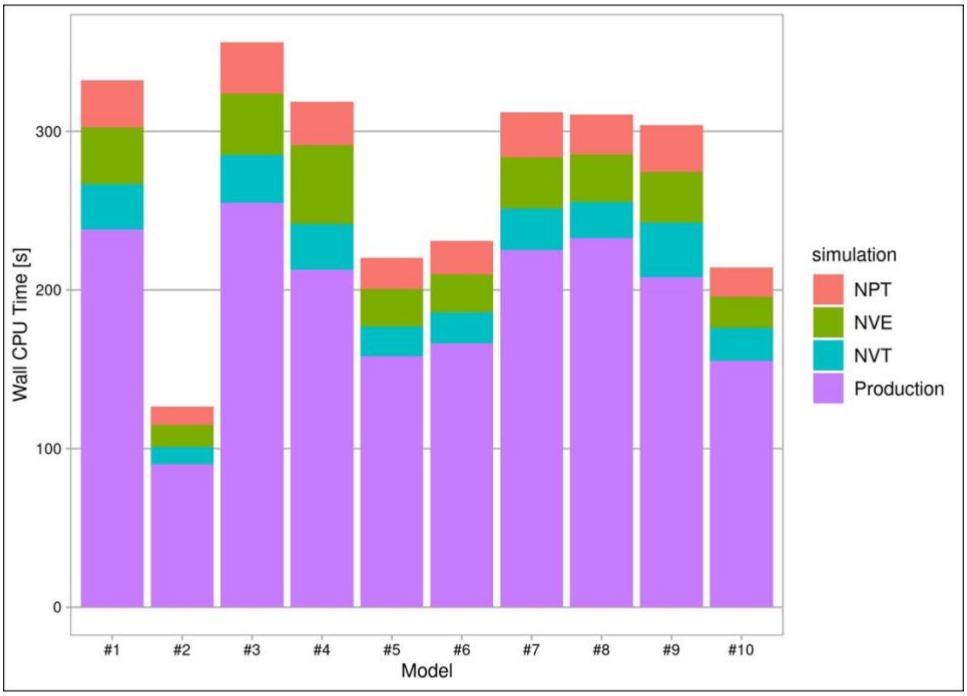

Figure 1. Wall CPU time to obtain the electrolyte model from a series of molecular dynamics simulations. The mean square displacement analyses are also included.

In comparison to another developed software parser, the present implementation is more user friendly and relatively straight forward to the properties of interest. As a comparison, ASE is known as one of the most powerful software to connect many quantum chemical and classical mechanics packages. However, the usage needs a high learning curve, at which the user will need to edit their own Python script to perform the simulations. Thus, there will be extra time and effort to write their own scripts.

3.2. Reaction spontaneity predictor at quantum mechanics level

The combustion reactions of alkanes in general are always spontaneous in nature. However, the trend of the reaction spontaneity as the function of *n* in C_nH_{2n+2} can be varied. Figure 2 shows the relation between the number of carbon atoms in the alkanes with the internal energies, enthalpy, Gibbs free energy, and entropy of combustion. It is clarified that the number of carbon atoms is proportional to all thermochemical properties. The chemical spontaneity, which is represented by the Gibbs free energy of combustion, increases as the increasing of the number of carbon atoms in the molecule. As shown by the blue and black lines, the enthalpies of combustion always have similar value to the internal energy of combustion. Thus, it is safe to assume that one can approximate the enthalpy of combustion by simply calculating the internal energy of combustion. Gibbs free energy, as shown by the green line, decreases as the decreasing number of carbon atoms.

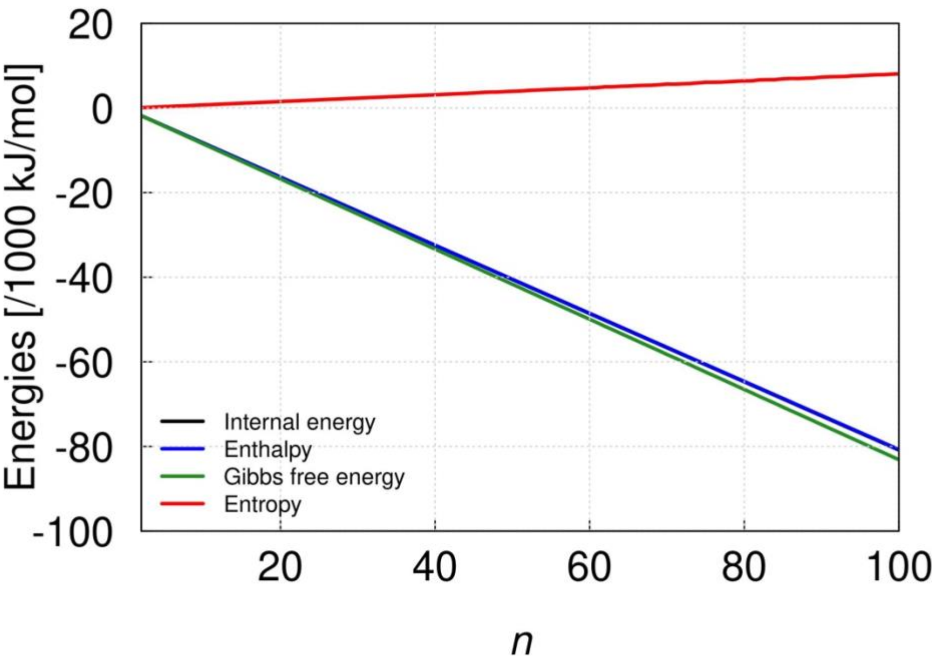

Figure 2. Calculated internal energy, enthalpy, Gibbs free energy, and entropy of combustion as the function of the number of carbon atoms (*n*).

As shown in Figure 2, it is also noticed that the entropy of combustion does not increase as significantly as the other energy terms. As the increasing of the number of carbon atoms, the entropy also increases, which leads to the deviation of the Gibbs free energy from the internal and enthalpy terms. Since the entropy becomes significantly high, namely up to 16 kJ/mol for C100H202, the Gibbs free energy becomes smaller by thousands of kJ/mol than the internal and enthalpy terms. The thermochemical properties of 100 alkane molecules as reported herein were obtained within a relatively short time. In principle, the analytical Hessian and geometry optimizations performed at DFTB took less than 1 minute. In the present case, hundreds of quantum chemical calculations only required 10 minutes to be completed. In contrast to this, one requires to manually create directories, making hundreds of alkane molecules, performing geometry optimizations, and finally calculating the Hessian matrices to get the thermochemical properties. Combining all of them together, approximately, one will require more than a day to perform such tasks.

4 Conclusion

The present work reported a simulation protocol to generate electrolyte solution and perform the analyses simultaneously by using a modular Python-based code, namely, the simulation protocol environment (SPE). The efficiency of the SPE code is shown by its ability to calculate in a relatively short time in a laptop or a personal computer. Designing electrolyte and predicting chemical reactions can be performed within minutes. Therefore, the SPE code would be useful for the experimental chemist who wants to predict the transport properties of electrolyte for battery applications and predict the spontaneity of the chemical reactions. The present code implementation can be accessed via a personal request. The Windows user may need to install Docker program to allow them to emulate the Linux environment.

Acknowledgments. The authors thank the Department of Chemistry, faculty of Science and Computer, Universitas Pertamina for providing discussions and support to the SPE code developer.

References

[1] Kelly H. The classical definistion of a pandemic is not elusive. Bulletin of the World Health Organization. 2011; 89:540-541.

[2] Khlystova O, Kalyuzhnova Y Belitski, M. The Impact of the COVID-19 pandemic on the creative industries: A literature review and future research agenda. Journal of Business Research. 2022;139:1192-1210.

[3] Meyer BH, Prescott B, Sheng XS. The impact of the COVID-19 pandemic on business expectations. International Journal of Forecasting. 2022;38:529-544.

[4] Abramo G, D'Angelo CA, Mele I. Impact of Covid-19 on research output by gender across countries. Scientometrics. 2022:1-16. Doi: https://doi.org/10.1007/s11192-021-04245-x.

[5] Riccaboni M, Verginer L. The impact of the COVID-19 pandemic on scientific research in the life sciences. PLoSONE. 2022;17.

[6] Hordosy R, McLean M. The future of the research and teaching nexus in a post-pandemic world. Educational Review. 2022;74(3):1-24.

[7] Mayzel A, Medvedev A, Temkina V, Pankova L, Sivolenko E. On the Experience of Using NI ELVIS III in Remote Laboratory Practice During Pandemic Lockdown. In: Velichko E, Kapralova B, Karasov P, Zavjaloc S, Angueira P, Andreev, editors. international youth conference on electronics, telecomunications and information technologies. Cham: Springer; 2022. p. 77-83.

[8] Abriata LA. How Technologies Assisted Science Learning at Home During the COVID-19 Pandemic. DNA and Cell Biology. 2022;41:19-24.

[9] Zarcone D, Saverino D. Online lessons of human anatomy: Experiences during the COVID-19 pandemic. Clinical Anatomy. 2022;35:121-128.

[10] Eilks I, Lathwesen C, Belova N. Students' perception of an individualized open lab learning experience during the COVID-19 pandemic in German Chemistry Teacher Education. Journal of Chemical Education. 2022;99(2):1628-1634.

[11] Hamilton NB, Remington JM, Schneebeli ST, Li J. Outcome-based redesign of physical chemistry laboratories during the COVID-19 Pandemic. Journal of Chemical Education. 2022;99:639-645.

[12] Guha R, Willighagen E, Zdrazil B, Jeliazkova N. What is the role of cheminformatics in a pandemic?. Journal of Cheminformatics 2021;13(16):1-3.

[13] Kuroki N, Mori H. Comprehensive physical chemistry learning based on blended learning: a new laboratory course. Journal of Chemical Education. 2021;98:3864-3870.

[14] Neese F. The ORCA program system. WIREs Computational Molecular Science. 2012;2(1):72-78.

[15] Neese F. Software update: The ORCA program system, version 4.0. WIREs Computational Molecular Science. 2017;8.

[16] Barca GM, Bertoni C, Carrington L, Datta D, Silva N, Deustua JE, et al. Recent developments in the general atomic and molecular electronic structure system. The Journal of Chemical Physics. 2020;152. Doi: https://doi.org/10.1063/5.0005188.

[17] Hourahine B, Aradi B, Blum V, Bonafe F, Buccheri A, Camacho C, et al. DFTB+, a software package for efficient approximate density functional theory based atomistic simulations. The Journal of Chemical Physics. 2020;152. Doi: https://doi.org/10.1063/1.5143190.

[18] Nishimura Y, Nakai H. DCDFTBMD: Divide-and-conquer density functional tight-binding program for huge-system quantum mechanical molecular dynamics simulations. Journal of Computational Chemistry. 2019;40:1538-1549.

[19] Bannwarth C, Caldeweyher E, Ehlert S, Hansen A, Pracht P, Seibert J, et al. Extended tight-binding quantum chemistry methods. WIREs Computational Molecular Science. 2021;11.

[20] Bannwarth C, Ehlert S, Grimme S. GFN2-xTB—An accurate and broadly parameterized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. Journal of Chemical Theory and Computation. 2019;15: 1652-1671.

[21] Berendsen HJ, Spoel D, Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. Computer Physics Communications. 1995;91:43-56.

[22] Lindahl E, Hess B, Spoel D. GROMACS 3.0: A package for molecular simulation and trajectory analysis. Molecular Modeling Annual. 2001;7:306-317.

[23] Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJ. GROMACS: Fast, flexible, and free. Journal of Computational Chemistry. 2005;26:1702-1718,

[24] Hess B, Kutzner C, Spoel D, Lindahl E. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. Journal of Chemical Theory and Computation. 2008;435-447.

[25] Pronk S, Pall S, Schulz R, Larsson P, Bjelkmar P, Apostolov R, Shirts MR, et al. GROMACS 4.5: A hgh-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics. 2013;29: 845-854.

[26] Pall S, Abraham MJ, Kutzner C, Hess B, Lindahl E. Tackling exascale software challenges in molecular dynamics simulations with GROMACS. In: Markidis S, Laure E, editors. Solving software challenges for exascale. EASC 2014. Lecture Notes in Computer Science. Cham: Springer; 2014. p. 3-27.

[27] Abraham MJ, Murtola T, Schulz R, Pall S, Smith JC, Hess B, et al. GROMACS: High performance molecular simulations through multilevel parallelism from laptops to supercomputers. SoftwareX. 2015;1-2:19-25.

[28] Larsen AH, Mortensen JJ, Blomqvist J, Castelli IE, Christensen, Dulak M, et al. The atomic simulation environment – A Python library for working with atoms. Journal of Physics: Condensed Matter. 2017;29.

[29] Sakti AW, Wahyudi ST, Ahmad F, Darmawan N, Hardhienata H, Alatas H, Effects of salt concentration on the water and ion self-diffusion coeffcients of a model aqueous sodium-ion battery electrolyte. The Journal of Physical Chemistry B. 2022;126(11):2256-2264.

[30] Okoshi M, Chou CP, Nakai H. Theoretical analysis of carrier ion diffusion in superconcentrated electrolyte solutions for sodium-ion batteries. The Journal of Physical Chemistry B. 2018;9:2600-2609.

[31] Wang J, Yamada Y, Sodeyama K, Chiang CH, Tateyama Y, Yamada A. Superconcentrated electrolytes for a high-voltage lithium-ion battery. Nature Communications. 2016;7.

[32] Yamada Y, Yamada, A. Review – Superconcentrated electrolytes for lithium batteries. Journal of The Electrochemical Society. 2015;162.

[33] Martinez L, Andrade R, Birgin EG, Martinez JM. Packmol: A package for building initial configurations for molecular dynamics simulations. Journal of Computational Chemistry. 2009;30:2157-2164.

[34] Martinez JM, Martinze L. Packing optimization for automated generation of complex system's initial configurations for molecular dynamics and docking. Journal of Computational Chemistry. 2003;24:819-825.

[35] Sprenger KG, Jaeger VW, Pfaendtner J. The General AMBER Force Field (GAFF) can accurately predict thermodynamic and transport properties of many ionic liquids. The Journal of Physical Chemistry B. 2015;119:5882-5895.

[36] Elstner M, Porezag D, Jungnickel G, Elsner J, Haugk M, Frauenheim T. Self-consistent charge density-functional tight-binding method for simulations of complex materials properties. Physical Review B. 1998;58.

[37] Gaus M, Cui Q, Elstner M. DFTB3: Extension of the self-consistent-charge density-functional tight-binding method (SCC-DFTB). Journal of Chemical Theory and Computation. 2011;7: 931-948.

[38] Caldeweyher E, Mewes JM, Ehlert S, Grimme S. Extension and evaluation of the D4 London-dispersion model for periodic systems. Physical Chemistry Chemical Physics. 2020;22:8499-8512.

[39] Caldeweyher E, Bannwarth C, Grimme S. Extension of the D3 dispersion coefficient model. The Journal of Chemical Physics. 2017;147.

[40] Wang J, Wang W, Kollman PA, Case DA. Automatic atom type and bond type perception in molecular mechanical calculations. Journal of Molecular Graphics and Modelling. 2006;25.

[41] Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA. Development and testing of a general AMBER force field. Journal of Computational Chemistry. 2004;25:1157-1174.

[42] Chou CP, Sakti AW, Nishimura Y, Nakai H. Development of divide-and-conquer density-functional tight-binding method for theoretical research on li-ion battery. The Chemical Record. 2018;18:1-13.

[43] Merkel D. Lightweight linux containers for consistent development and deployment. Linux journal. 2017;239.