GridMol2.0: Implementation and Application of Linear-scale Quantum Mechanics Methods and Molecular Visualization

Baohua Zhang[1], Yingjin Ma[1], Xinsheng Jin[2], Ying Wang[2], Bingbing Suo[3], Xiao He[2], and Zhong Jin[1]

[1]Computer Network Information Center Chinese Academy of Sciences
[2]East China Normal University
[3]Institute of Modern Physics

March 11, 2020

Abstract

GridMol is a “one-stop” platform for molecular modeling, scientific computing and molecular visualization aided by High Performance Computing Environment. GridMol version 2.0 emphatically introduces two unique features, the first is fragment-based linear scaling quantum chemistry methods, such as molecular fractionation with conjugate caps and fragment molecular orbital methods; the second is visualization of computational processes, such as structural optimization and intrinsic reaction coordinate calculation. Compared with version 1.0, fragment-based linear scaling quantum chemistry methods implemented in GridMol version 2.0 can be used as a useful tool for performing quantum calculations for large molecular systems to explore the mechanisms involved in protein–ligand or targeted-drug interactions.

Introduction

GridMol is designed as a grid application for molecular modeling and visualization.[1] It comprises a typical browser and server platform based on Java and Java3D, which have “write once and run anywhere” capability.[2,3] User’s computing tasks can be submitted to high-performance computers (HPCs) as jobs via China National Grid (CNGrid)[4,5] or Scientific Computing Grid (ScGrid)[6], which provides unified high performance computing services based multiple heterogeneous HPCs. CNGrid has aggregated more than 460 petaflops of computing capability and 310 million gigabytes of storage from 19 supercomputing sites in China, including Sunway TaihuLight (No. 3 in 2019 HPC TOP500 list) and Tianhe-2A (No. 5 in 2019 HPC TOP500 list), and etc. [7]ScGrid has aggregated more than 215 petaflops of computing capability from 10 institutes of the Chinese Academy of Sciences (CAS). Both of CNGrid and ScGrid provide RESTful APIs to provide services, including unified user authentication and authorization, job submitting and checking, secure user data management, and *etc.* . Based on the grid environment, GridMol focuses on providing ”one-stop” computational chemistry solutions, including molecular modeling, visualization, animation, job submission, management, and results analysis.[1,8]

The main features of GridMol version 1.0 include molecular modeling, visualization, and job submission.[1,9] GridMol allows all basic molecular modeling functions, including modification of bond length, angle, and dihedral angle, as well as addition or modification of atoms and radicals. Further, users can use GridMol to read files of common chemistry software formats, such as PDB, MOL2, GJF, and XYZ, and visualize three-dimensional (3D) structures via different display formats, including tube, ribbon, and cartoon models, for large molecules and other formats, such as line, ball-and-stick, and space-filling models, for molecules.[3] GridMol allows easy access for users (those without an account can register for free through http://www.cngrid.org/) to launch a job on remote HPCs. While the job runs, GridMol monitors its status and provides users with the ability to terminate the job via the RESTful APIs provided by CNGrid or ScGrid. After several years of development, GridMol now represents a powerful research tool.

GridMol version 2.0 introduces new and unique implementations, the most important of which are fragment-based linear scaling quantum chemistry methods (FLSMs) and grid-based interactive visualization. FLSMs offer a unique solution for calculations involving large molecules by decomposing a large molecular system into subsystems that can be calculated at quantum mechanical (QM) levels, as well as representing properties of intractable super systems as a reassembly of individual fragments.[10-13] Among many FLSMs, molecular fractionation with conjugate caps (MFCC)[14] and fragment molecular orbital (FMO)[15] methods (both described later) are selected for implementation in GridMol. This is important for generalized automation of fragment-based calculation for the following reasons:

1) it addresses computational limitations associated with performing electrostatic calculations on large molecules by allowing an accurate, automated fragmentation step to do something cumbersome and inaccurate when done manually;

2) it increases the ease of input-file preparation for QM procedures by enabling performance of FLSMs, even without expert knowledge; and

3) it allows users to conveniently utilize HPCs to speed up the computational process, especially given that substructures can be treated separately through parallel calculations; thus, the computational time for fragments is ~equivalent to that required for the largest fragment.

Another important feature is grid-based interactive visualization. When performing molecular calculations, such as structure optimization (OPT) and intrinsic reaction coordinate (IRC) calculation, which involve multiple self-consistent steps, users usually need to check intermediate results in order to examine structure and/or energy evolution, especially when establishing new systems.[16-18]Monitoring the calculation in real time allows users to identify problems while not wasting computing time or resources. Although graphical tools exist for analyzing the results calculated by chemistry software (e.g., GaussView for Gaussian, VMD for NAMD, MS Visualizer for Materials Studio),[19-21] these are desktop applications, unable to connect to several remote HPCs via simple APIs rather than through open ports. Although GridChem can launch and monitor calculations on supercomputers from remote sites, it is still a desktop application.[22] GridMol is a web-based application that can securely launch jobs on multiple remote HPCs via via the RESTful APIs. Additionally, it is difficult to visualize results on HPCs directly, so process monitoring is necessary for visualizing intermediate computational results. Due to the cross-platform properties and the simple API associated with CNGrid or ScGrid enviroment, process monitoring is convenient for different remote HPCs with minimal lag time and enables identification of simulation problems in time to either terminate or restart a job.

Moreover, other minor updates, including interfaces for input-file preparation for chemistry software and extensive result analysis, have been introduced. In this paper, we provide two practical examples in order to illustrate FLSM applications in GridMol. For MFCC, we assessed the performance of different density functional theory (DFT) methods for calculating ligand–protein-binding energies. Furthermore, we used FMO, combined with transition-state (TS) calculation, to provide a new solution for studying the dissociation mechanism related to ligand-targeted drugs (LTDs).[23]

Theoretical Background

Fragment-based Linear Scaling Computation Methods

QM methods have been widely applied in chemistry because they can evaluate electron interactions relative to molecular mechanical (MM) methods. However, due to the steep computational scaling associated with system size, it is difficult, or even impossible, to perform quantum calculations for large molecular systems, such as biomolecules that contain hundreds or thousands of atoms. The desire to study systems larger than what was computationally feasible led to the development of novel methods, including QM/MM methods, semi-empirical approaches, and reduced-scaling methods.[11] Another method is linear scaling,[24] a thriving field of research into efficient calculation of large molecular systems using only modest requirements for memory and CPU time. Some linear scaling methods rely on screening or approximating electron-repulsive integrals,[25-30]such as the fast multipole method[25] and linear scaling exchange[29]; however, they do not split a whole molecule into fragments, and can only achieve linear scaling for one-dimensional systems, such as alkane chains. In contrast, FLSMs as another important class of scaling method can achieve real linear scaling for 3D systems, such as proteins.[10] These represent examples of the significant progress made in developing and applying new fragmentation methods.[31-40]

Li et al.[32] grouped FLSMs into density- and energy-based methods. In the present study, we divided all FLSMs into 'overlapping' and 'disjoint' methods according to fragment formation. FLSMs, such as MFCC, FMO, generalized energy-based fragmentation (GEBF), molecular-tailoring approach (MTA), kernel energy method, X-Pol (previously referred to as MODEL), use strictly linear scaling and have received increased attention.[10,14,15,32,41-48] For example, FMO was used by Heifetz et al.[49,50] to investigate agonist–orexin-2 receptor interactions and optimize interleukin-2-inducible T cell kinase inhibitors. Additionally, MFCC was used by Liu et al.[51] for geometry optimization and vibrational spectrum calculation of proteins, and Singh et al.[52] used the MTA method to estimate binding energies for large water clusters.

Performing an FLSM calculation usually requires three steps. The first step is dividing the large molecular system into subsystems, which might or might not contain buffer atoms, according to different methods. Ideally, the correct fragmentation operation ensures the local interaction of every fragment. Second, input file(s) of appropriate quantum chemistry software should be prepared and used for subsystem calculation. The final step is assembling the calculation results of substructures to obtain the original system's properties, such as charge, energy, or energy gradient. Correct and efficient performance of FLSM calculations is not straightforward, especially for the molecule fragmentation step, which is cumbersome. Combining the three steps into a single, automated solution in one platform or software package would significantly lower the barrier of using FLSMs.

Among FLSMs, two methods typically belong to different subclasses, with MFCC and FMO chosen for implementation in this study. MFCC was proposed by Zhang *et al.* in 2003[14] and represents an inclusion-exclusion principle-based method that belongs to the 'overlapping' FLSM subclass. It is ideally suited for calculations involving large biomolecules, such as ligand–protein-binding energies. There are other methods, including generalized (G)MFCC/MM and electrostatically embedded (EE)-GMFCC,[14,53,54]developed based on the MFCC method; however, a platform or software suite for simplified use of the MFCC method for research is currently unavailable. In this study, we constructed an automated process to make MFCC a useful tool, especially for users unfamiliar with such methods.

Kitaura and co-workers originally proposed the FMO method, which belongs to the 'disjoint' FLSM subclass, in 1999 to calculate the energies of large molecular systems and with properties obtained from many-body expansion or FMO calculations.[15,55-59] FMO is a well-established tool for calculating energies and other properties, optimizing structure and study time evolution with molecular dynamics (MD), and investigating interactions in large molecular systems.[10,57,60] Several improved methods have been developed based on FMO, including effective fragment-potential FMO[61], FMO/polarizable continuum model (PCM)[62], and FMO-long-range correction density functional tight binding.[63] Additionally, FMO has been implemented in several programs, including General Atomic and Molecular Electronic Structure System [GAMESS (US)], ABINIT-MP, OpenFMO, and parallelized *ab initio* calculation system (PAICS).[64-67] GAMESS (US) incorporates a majority of FMO-related methods and is a widely used package for FMO research.[57] OpenFMO is an open-architecture program targeting effective FMO calculations on massive parallel computers, especially GPU-accelerated computers.[67] The availability of graphical user interfaces makes FMO application relatively easy for preparing calculations and visualizing results.[34,40,68] For example, FragIt is used to prepare input files for FMO calculation in GAMESS, but it cannot use HPCs to accelerate the computing process, and it includes limited results analysis ability.[35] Additionally, Facio is used for FMO input-file preparation for PC-GAMESS; however, its implementation as a Windows application restricts its use.[69] BioStation Viewer and

PAICSView are user interfaces for ABINIT-MP and PAICS, respectively, and both have limited abilities to interact with HPCs. In the present study, we implemented the full FMO method of GAMESS into GridMol to allow users to prepare FMO input files easily and use HPCs to accelerate the computation process.

Program Details

Grid-Based Implementation on FLSMs

In our previous work, we proposed a general macromolecular automatic fragmentation method.[70] In the present study, we implemented the complete processes of FLSMs in GridMol. Figure 1 shows the process of user selection of a specific method, calculation parameters, and the appropriate QM procedure, with the remaining jobs finished by GridMol and remote HPCs through implementing two specific methods (MFCC and FMO). The flowchart in Figure 2 shows the steps involved in the process.

Figure 1. System schematic of FLSMs implemented in GridMol.

Figure 2. Flowchart of FLSM calculation in GridMol.

Step 1: Automation of the Partitioning Schemes

Automatic fragmentation of a large molecule is desirable, because of the time-consuming and error-prone aspects associated with manual fragmentation. This implementation supports protein files in PDB or MOL2 format.

For MFCC, splitting the buried residues of the protein results in two incomplete points that require two groups of conjugated caps (‘caps’) to cap the severed valences (Figure 3). For example, if the split incomplete fragments include three amino acids, the estimated atoms in the subsystems and two caps are always >60, even for small residues, constituting a relatively large system for QM calculations. According to computational efficiency and convenience, each fragment was generated by breaking the peptide bond. Assuming that one protein contains N amino acid residues (Figure 3; N = 3), N fragments and N - 1 caps will be generated after fragmentation. First, all molecular coordinates are scanned to search the split points, after which the protein is split into N incomplete fragments; each includes one or two caps according to residue location (terminus or buried area). Splitting buried residues results in two incomplete points, which require two caps (e.g., Frag 2 in Figure 3). For residues located at termini, there is only one split point, requiring only one cap (e.g., Frags 1 and 3 in Figure 3). The caps are not only used as buffered groups for residues, but also mimic the protein’s real environment.

Figure 3. Graphical representation of MFCC fragmentation of three residues in a protein.

Figure 4. Graphical representation of FMO-based fragmentation results.

For the FMO method, charge partitioning makes auto fragmentation difficult. Because exchange and self-consistency are local in most molecules, non-local regions are treated using the coulomb operator, and molecular calculations are performed individually within the overall system’s coulomb field. The coulomb bath allows for fragmentation without hydrogen capping. For proteins, fragments can be generated automatically by breaking the Cα–C(O) bond. After fragmentation (Figure 4), a specific number of electrons are assigned to each fragment according to bond-attached atom, bond-detached atom, and DUMMY, which mimics the coulomb field. The fragments’ molecular orbitals and the contributions from pairs of fragments are calculated to account for the molecule’s total energy.

Step 2: Preparation and Submission of Input Files for the QM Procedure

For MFCC, after generating fragments and conjugating caps, all fragments (–ligand), caps (–ligand), and ligand molecules are evaluated by QM calculations (if N fragments have been generated for a protein, the number of QM input files will be 4N – 1). Because the first (last) fragment and conjugated cap(s) share the same structure, they are not calculated, and the number of input files is 4N - 9 (large, even for a moderately sized protein). Therefore, automated preparation of QM input files is desirable. In the implementation of

GridMol version 2.0, users can select one QM package (e.g., Gaussian) and provide parameters to allow automatic generation of the input files.

For FMO, because the process of generating a GAMESS input file is not straightforward, users must create a complex input file while consulting a manual. In the implementation described here, we analyze each fragment's characteristics and perform an automated generation of the input file. Bonds are fractioned electrostatically, and bond-specific electron pairs remain intact. Each atom in a molecule is grouped into a unique fragment, with the charge of each fragment assigned according to its electrons, which are assigned heterolytically during molecule fragmentation. In FMO, the sp^2 or sp^3 hybrid orbitals are used for bond detachment. In this study, we created orbitals for sp^3 C, sp^3 N, and sp^2 C for frequently used Gaussian-based functions [STO-3G, 6-31G, 6-31G (d), and 6-31G (d, p)] and inserted them into the database. After selecting the basis set, the hybrid molecular orbitals block for a specific basis set can be automatically written into the $FMOHYB group of the input file.

Once generated, the QM input files for MFCC, or GAMESS for FMO, can be submitted to the grid environment, where a 'best' destination node (according to the scheduling policy) is selected to finish the job. The input files can be processed concurrently, with each activity accelerated by parallel computing.

Step 3: Analysis of QM-calculation Results

After the calculation process has finished and the result files generated for MFCC, each result file is analyzed by GridMol automatically to examine and assemble the results to obtain the assembled system's properties (e.g., energy). The total energy is approximated using Eq. (1):

$$E = \sum_{i}^{N} \left(E_{fragi-lig} - E_{\mathrm{fragi}} - E_{\mathrm{lig}}\right) - \sum_{i=1}^{N-1} \left(E_{capi-lig} - E_{\mathrm{capi}} - E_{\mathrm{lig}}\right)(1)$$

where E is the interaction energy between large molecules (e.g., a protein and a ligand), $E_{\text{fragi-lig}}$, E_{fragi}, and E_{lig} are the energy of fragment-ligand molecules and fragments and ligands, respectively, and $E_{\text{capi-lig}}$, E_{capi}, and E_{lig} represent the contributions of caps and ligands.

For FMO, the total energy of the system can be written as eq. (2)

$$E = \sum_{i=1}^{N} E_i + \sum_{i>j}^{N} \left(E_{\mathrm{ij}} - E_i - E_j\right) + \sum_{i>j>k}^{N} \{(E_{\mathrm{ijk}} - E_i - E_j - E_k) - (E_{\mathrm{ij}} - E_i - E_j) - (E_{\mathrm{jk}} - E_j - E_k) - (E_{\mathrm{ki}} - E_k - E_i)\} + \ldots(2)$$

where E_{i}, E_{ij}, E_{ijk}represent monomer, dimer, and trimer energies, which can be obtained from GAMESS output file for FMO2 and FMO3 calculations.

Grid-based Molecule Visualization

We implemented the interactive visualization process OPT and IRC calculation of the Gaussian method in GridMol based on our previous work.[71] In the grid environment, users with a license for Gaussian can submit jobs to HPCs (the grid administrator can provide usage support). When a Gaussian OPT or IRC job is submitted to the grid environment, grid-based visualization allows users to view the calculation process (Figure 5). The contents of the output file are extracted through the 'grid module', which uses RESTful APIs to obtain data from remote HPCs. The 'data processing module' is used to analyze the data and provide the processed data to the 'GridMol graphical interface' for visualization. The 'Grid module' uses a Model-View-Control model, the advantage of which is its separation of the user interface from the data, and a controller responsible for controlling the model and the view. To reduce client load and strengthen the support for cross-platform and distributed computing, we adopted remote method invocation technology to communicate between client and server.[72]

Figure 5. System design of interactive visualization in GridMol.

The processes associated with rapid visualization and updating of changes made to 3D molecules, as well as how to ensure frequent data acquisition and avoid additional system burden, represent important fields of research. For molecular visualization, we used the scene graph, which is separated from the molecular structure and provides common characteristics, such as background, appearance, and molecule behavior, whereas the molecular structure uses atoms and bonds to allow zooming in/out and translation. This separation eliminates the need to repaint the total molecule and change the scene graph by simply translating atoms and redrawing the bonds, thereby significantly improving display efficiency.

To ensure frequent data acquisition, we set the data acquisition time interval to 10 seconds based on experimental analyses. Figure 6 illustrates the rule of incremental data-transfer from the breakpoint. During every data-checking activity, useful data are saved locally and a breakpoint established. When users check the calculation process, data are downloaded from only the breakpoint, thereby greatly reducing the burden on the grid and improving real time system performance.

Figure 6. Sketch map of continuous transfer of data from breakpoints.

Benchmarks

MFCC

To verify the code's validity, we performed an interaction-energy calculation for a peptide with six residues and a water molecule using MFCC and full-system computation, respectively. Figure 7 shows the structures with different relative positions of the peptide and water molecule selected. The water molecule moved along the X-axis at intervals of 0.5 Å.

Figure 7. Structures of peptides (cube representations) and a water molecule (ball-and-stick representation).

The entire process, including 1) fragmentation, 2) generation of Gaussian input files and job submission, and 3) results analysis, is illustrated in Figure 8. For fragment energy calculation, we used the B3LYP functional together with the 6-31G(d) basis. The input files were submitted to the grid environment, and a grid node (ERA, located at the Computer Network Information Center, CAS, China) installed with the Gaussian 09 package was selected to execute QM calculations.[73]

Figure 8. MFCC computational process in GridMol. a. Fragmentation and preparation of input files, b. job submission to the grid, c. job management, and d. results analysis.

As seen from the calculated results (Figure 9), when the distance between the protein centroid and the water centroid is >8.5 Å, the interaction energy becomes negative, indicating an attractive protein–ligand interaction (where the ligand stabilizes the protein structure) and the most stable structure at a distance of 9 Å based on the lowest interaction energy. The relative errors between fragment-based and full-system calculations were <3% for all the calculations, suggesting the validity of MFCC implementation in GridMol.

Figure 9. Chart showing the interaction energies from full-system and MFCC computations. Distance is between the peptide centroid and the water centroid.

FMO

Here, we used the FMO method to compute the binding energy between the Trp-cage protein (PDB ID: 1L2Y; 20 residues with 304 atoms using the structure of the first isomer in the PDB file)[74] and a water molecule, which was placed in the protein's binding pocket. The protein–water complex is shown in Figure 10. The initial partial charges were obtained using the antechamber tool at the am1bcc level, and atom types in the general AMBER force field were assigned to the molecule,[75,76] after which topology files were generated using LEaP in the antechamber tool.

We used the AMBER 2016[77] package, using 200 steps of minimization for the complex or separate ligand/protein, with the first 50 steps involving use of the steepest descent algorithm, followed by implementing the conjugate-gradient algorithm and then GridMol for fragment calculation. The FMO method and full-system computation were performed for comparison. Figure 11 shows the FMO process in GridMol, and a restricted Hartree–Fock method together with 6-31G(d) was used for single-point calculations. Full-system and FMO results were -0.446 and -0.453 kcal/mol, respectively, with a relative error of 1.6 %, which was within a reasonable range.

Figure 10. The Trp-cage protein–water complex.

Figure 11. The FMO process in GridMol

Grid-based Visualization of Changes in Molecular Structure

We selected the first fragment of the 3FLY protein to perform optimization calculations using the Gaussian 09 package (Figure 12a). The panel for input-file creation was used to initialize the input parameters and preview or modify the file. After providing the necessary information, including job name, CPU cores, and wall time, the RESTful APIs can be used to dispatched the job to the appropriate queue. During runtime, it is possible to use the visualization panel (Figure 12b).

The visualization panel shows the structure and the optimization plot, which can be checked at any time while the job is running. Animation and multiple view (Figure 12b-3) options allow different visualizations of the molecule structure. Optimization plots, including energy and root-mean-square curves, are shown at the right panel, and in the lower panels, the console records the processes associated with data processing (Figure 12b-1), whereas the molecule panel shows the geometry information of the structure, and the criterion panel displays the minimum optimization convergence criteria of the Gaussian optimization calculation (Figure 12b-2). Initial checking of the calculations downloads a results file from the HPCs for parsing by GridMol. The visualization panel displays the 3D structure and associated optimization curves, along with data benchmarks for use during subsequent rounds of optimization.

Figure 12. Grid-based visualization of the optimization process.

Similar to the optimization process, the structure and curves from the IRC calculation can also be viewed in real time. An example proton-transfer process from the C of the carboxyl group to the N of the amide group in glycine (Figure 13) reveals the TS as the starting point of the IRC calculation (step size, 0.06 Bohr; maximum number of steps taken in each trajectory, 10). The two trajectories' final structures were similar to those for the reactant and product, respectively.

Figure 13. Grid-based visualization of the IRC process.

Practical Applications

MFCC

We used different QM methods to calculate binding free energy for two complex systems.[78] As shown in Figure 14, the complex systems comprised a protein built according to a crystal structure (PDB ID: 3FLY) and two different ligands (PDB IDs: 3FMH and 3FLZ). First, the initial partial charges are obtained using the antechamber tool at the am1bcc level, and the atom types in GAFF are assigned to the two molecules. LEaP is then used for rapid generation of topology files for use with the AMBER simulation. The structures are minimized initially, using steepest descent for the first 50 cycles and then conjugate gradient for 150 cycles with an implicit solvent model. The systems are then gradually heated 100 K every 5 ps from 50 K to 300 K, after which the systems are equilibrated for 10 ps. We then collected an MD simulation for 1 ns with a 1-fs time step, and GridMol was used to load the relaxed system and perform autofragmentation. We used different QM methods, together with the 6-31G(d,p) basis set in the Gaussian 16 package, to calculate the energy of each fragment, cap, and ligand. Additionally, we used the AMBER ff9SB force field to calculate the energy of the protein–ligand complex and the protein and ligand separately.

we conducted a comparative study to investigate the performance of widely used quantum mechanical methods in the treatment of the protein and ligands, including available functionals representing hybrid GGA (B3LYP and PBE0), meta-GGA (TPSS), hybrid meta-GGA (M06 and M06-2X), range-separated hybrid GGA (CAM-B3LYP and ωB97XD), global-hybrid meta-NGA (MN15) classes, and the MP2 method.[79-85]

The interaction energies calculated by the AMBER ff99SB force field for 3FMH and 3FLZ were -122.99 kcal/mol and -38.99 kcal/mol, respectively. Table 1 shows the MFCC calculation results, which were almost the same order of magnitude as the AMBER force field. This suggested that the fragmentation steps were correct for the two complexes. We used the calculated results for the MP2 method as a reference of reliability. The complex, with 5640 atoms, was fragmented into 1387 fragments, with the number of atoms in many of the fragments >70 and with inter-molecular distances >10 Å. We used DFT-D3 with zero damping[86-88] for dispersion correction for all functionals except the ωB97XD method, which already includes empirical dispersion correction.

Figure 14. The protein–ligand complex (PDB ID: 3FLY). a. p38a_3fmh and b. p38a_3flz.

The interaction energies calculated, by DFT functionals, ranged from -88.98 kcal/mol to -110.73 kcal/mol for 3FMH and -82.14 kcal/mol to -104.41 kcal/mol for 3FLZ. These results are consistent with those of MP2, indicating the importance of functional calculation with D3 dispersion correction. Among the DFT functionals, the results for higher rungs of Jacob's ladder showed lower interaction energies than those for the lower rungs, such as TPSS versus B3LYP. The results using D3 correction showed improvements in describing weak interactions. MP2 is more computationally expensive than DFT; however, the energy calculated by MP2 was similar to that by DFT-D3 (such as M06-2x). Therefore, these results suggested that DFT with dispersion correction is the better choice for efficient calculation of interaction energy.

Table 2. Parallel computing performance for 3FMH

Nodes/Processes/Threads	Cores	Time (s)	Parallel efficiency [a]	Parallel efficiency [b]
32/2/12	768	1938	—	—
64/2/12	1536	1346	71%	—
128/2/12	3072	922	52%	—

256/2/12	6144	798	30%	—
512/2/12	12288	448	27%	—
1024/2/12	24576	266	22%	84%
1536/2/12	36864	195	20%	77%
2048/2/12	49152	213	14%	53%

[a] Parallel efficiency relative to the result of 1008 cores.

[b] Parallel efficiency relative to the result of 12288 core

Massively Parallel Computing

For calculations made in the previous section, the input files for the fragments were submitted to one specified queue, and each corresponded to an independent job ID. Additionally, it is possible to submit MFCC jobs by interfacing with the LSSCF module of BDF[42,44,86-88] and X-Pol.[42] In the present study, we used GridMol version 2.0 to supply the fragmentation scheme (e.g., monomers and metal-ligand dimers) and then interfaced with the X-Pol package for massively parallel calculations using the LSSCF module in the BDF package. This allowed the hybrid MPI/OpenMP scheme to be used for accelerating the calculations according to optimized load-balance and task-tracking. Due to limited computing resources, we selected only part of 3FMH system, including 1020 atoms, for testing using PBE and STO-3G as the functional and basis set, respectively. Table 2 shows the acceleration ratio of 49,152 cores relative to 12,288 cores at 53%, indicating the fragment-based method as a better choice for large molecule systems according to accuracy and efficiency.

FMO

Safety and efficacy are important in precision medicine. An attractive strategy to maximize these areas is using LTDs, which typically comprise a targeting ligand, spacer, cleavable linker, and therapeutic payload.[23] Because LTDs can achieve the required therapeutic potency with minimal toxicity, the successful design of each component and an appropriate delivery mechanism remain active areas of research.[89-92] In the present study, we used the FMO utility in GridMol version 2.0 combined with TS theory to elucidate the releasing mechanism of a drug conjugate with a triple payload of paclitaxel. We focused only on the initial step (breaking the spacer–linker bond) in the drug-release process (Figure 15). First, the chemical structure of the ligand-AB_3 dendritic-prodrug conjugate (Figure 16) was prepared using the molecule-building module of GridMol. Seven components corresponded to four function types (i.e., targeting ligand, spacer, linker, and drug) (Figure 16).

Figure 15. Mechanism of drug release from the ligand-AB_3dendritic-prodrug conjugate.

Figure 16. The chemical structure of the ligand-AB_3dendritic-prodrug conjugate comprising seven components corresponding to four function types.

We performed geometry optimization at the PM6 level, using the PCM of solvation, in the Gaussian 09 package. We then elongated the CO–N(H) bond between components two and three and performed TS optimization using QST3 with the PCM at the same level as the optimization step. Atoms except for those related to the CO–N(H) of components two and three were frozen during transitions-state optimization for simplification because a more elaborate mechanism is beyond the scope of this study. Vibrational-mode analysis indicated a unique imaginary frequency of -344.76 cm^{-1} related to bond stretching along the direction of targeted CO–N(H) bond. IRC calculations to confirm the TS structure using a 0.2-Bohr step size showed that the structure (Figure 17) demonstrated a final point of forward direction (R') located at a C–N(H) bond distance of 1.68 Å, with a PM6 total energy of -0.575206 Hartree. To identify the remaining structural and energetic differences between the R' and the true-minimum R, we optimized the structure with all atoms relaxed in order to reach the ground state. The re-optimization of R' resulted in a fully optimized structure with a total energy of -0.676 Hartree and a CO–N(H) bond distance of 1.39 A, which was highly similar to the reactant and confirmed the correctness of desired TS.

Figure 17. IRC calculations of the ligand-AB_3dendritic-prodrug conjugate.

To investigate changes in interaction energy between various functional components along the IRC path, we performed GAMESS/FMO calculations at the M06-2x/6-31+G(d) level. FMO2 calculations were performed for the seven fragments corresponding to seven functional components in order to obtain four snapshots along the PES of the R' state to the TS. Figure 18 shows the interacting/bonding energies of fragments 1-2 and 3-4 remained similar along the reaction path (˜-30.0 Hartree), although the bonding energies of the linker–drug fragments (5-4, 5-3, 6-4, 6-5, 7-3, and 7-4) showed relatively large fluctuations at different stages of the reaction pathway. For example, the interacting/bonding energies of fragments 7-3 and 7-4 were -31.61 Hartree and -15.18 Hartree in R' (Figure 18a) and -0.01 Hartree and -52.89 Hartree at the IRC path's middle point (Figure 18b). For the spacer–linker fragment (2-3), the interacting/bonding energies were -29.36 Hartree in R' (Figure 18a) and -16.16 Hartree during the TS (Figure 18d). These changes indicated that the drug components in the ligand-AB_3 dendritic-prodrug conjugate were more flexible and experienced various structural changes relative to the other components, such as the ligand and spacer. Furthermore, the C–N(H) bond of spacer–linker fragment 2-3 can undergo heterolytic dissociation in the presence of a catalyst. According to a previous study,[89] enzymatic cleavage of the bond at Phe–Lys interaction can trigger dissociation of the conjugate and promote triple elimination to release the three paclitaxel molecules from each dendritic platform. Although full-stack calculations have not yet been performed for the entire therapeutic release mechanism, FMO, combined with TS calculations, represent a useful tool for studying the drug-release mechanism.

Figure 18. Structures (from R' to TS) and interaction heatmaps along the IRC path of the ligand-AB_3 dendritic-prodrug conjugate. a) R', b) the middle point1 of the IRC path, c) the middle point2 of the IRC path, and d) the TS. Positions of the selected points are shown in the IRC path sketch map (left).

Other Features Added to GridMol Version 2.0

1. Provide input-file preparation interfaces for computational programs, such as Gaussian, ADF, Molpro, NWChem, LAMMPS, and Gromacs (Figure 19).[92-96]. Simple, advanced, and expert input-file preparation interfaces have been developed for Gaussian software.
2. Smiles input-file formats can conveniently transfer between different file formats.
3. Results analysis for the Gaussian package, including job summary, charge distribution, spectra, and normal-mode vibrational analysis (Figure 20).

Figure 19. Job submission wizards used in software packages. a. Basic and b. advanced examples for wizards used in other software and the Gaussian package, respectively.

Figure 20. Results analysis in the Gaussian package a. Job summary, b. charge analysis, c. vibrational-mode analysis, and d. analysis of the infrared spectrum.

Conclusions

Here, we described several new features of GridMol version 2.0, including grid-based visualized implementation of FLSMs (MFCC and FMO) and visual analysis of OPT and IRC calculations. We demonstrated its effectiveness using benchmark examples and assessed DFT (with empirical dispersion correction) results using two protein–ligand complexes for comparison of different DFT methods for MFCC. Parallel-scalability tests indicated that MFCC interfaced with the LSSCF module in BDF and X-Pol achieved a high level of efficiency. GAMESS/FMO, combined with the TS method, was used to illustrate the dissociation mechanism for ligand-AB_3 LTDs. The findings confirmed that this version of GridMol is easily accessible to non-experts hoping to augment their experimental results with computational insights. However, further research to expand analysis of reaction mechanisms associated with LTDs, such as MD simulations combined with the FMO method, is desirable. Future GridMol implementations should consider including other FLSM methods, such as EE-GMFCC, GEBF, and the ability for 3D visualization (e.g., molecular orbital visualization). Furthermore, due to defects in the JAVA applet, the HTML-based web page requires reconstruction to offer a better user experience.

Acknowledgments

B. Zhang, Y. Ma and Z. Jin is thankful for the support from Ministry of Science and Technology of China (NO.2017YFB0202202), National Natural Science Foundation of China (NO. 21703260), Informatization Program of the Chinese Academy of Sciences (NO. XXH13506-403) and China National Grid (CNGrid), Scientific Computing Grid (SCGrid and) the National Supercomputer Center in Guangzhou.

X. Jin, Y. Wang and X. He is thankful for the support from Ministry of Science and Technology of China (No. 2016YFA0501700), National Natural Science Foundation of China (No. 21922301, 21673074 and 21761132022), Shanghai Municipal Natural Science Foundation (Grant No. 18ZR1412600), Youth Top-Notch Talent Support Program of Shanghai, and the NYU-ECNU Center for Computational Chemistry at NYU Shanghai. and the Supercomputer Center of East China Normal University (ECNU Multifunctional Platform for Innovation 001) for providing access to HPC resources.

B. Suo is thankful for the support from National Natural Science Foundation of China (No. 21673174 and No. 21873077).

Keywords: linear scaling method; MFCC; FMO; visualization analysis; LTDs

References and Notes

1. Sun, Y.; Shen, B.; Lu, Z.; Jin, Z.; Chi, X. *Journal of computer-aided molecular design* **2008** , *22(2)* , 119-129.

2. Kramer, D. *J White Paper, Sun Microsystems, Mountain View, CA***1996** .

3. Sowizral, H. A.; Deering, M. F. *IEEE Computer Graphics Applications* **1999** , *19(3)* , 12-15.

4. Xiao, H.; Wu, H.; Chi, X. In: Vicat-Blanc Primet P, Kudoh T, Mambretti J (eds) Networks for Grid Applications GridNets, Berlin, Heidelberg, 2009.

5. *http://www.cngrid.org(Access at Sep.30,2019*)

6.*http://cscgrid.cas.cn/.(Access at Sep.30,2019)*

7. *https://www.top500.org.*

(Access at Dec.30, 2019)

8. Ouyang, L.; Sun, Y.; Liu, J.; Jin, Z.; Lu, Z.; Chi, X. *Computer Engineering,China* **2009** , *35(20)* , 242-245.

9. Duan, Q.; Jin, Z.; Liu, Q.; Chi, X. *Information Computing and Applications* **2010** , 40-47.

10. Gordon, M. S.; Fedorov, D. G.; Pruitt, S. R.; Slipchenko, L.*Chemical Reviews* **2012** , *112(1)* , 632-672.

11. Alkimov, A. V.; V.Prezhdo, O. *Chemical Reviews* **2015** ,*115(12)* , 5797-5890.

12. Herbert, J. M. *J Chem Phys* **2019** , *151* , 170901.

13. Herbert, J. M. *The Journal of Chemical Physics* **2019** ,*151* , 170901.

14. Zhang, D.; Zhang, J. Z. H. *Journal of Chemical Physics***2003** , *119(7)* , 3599-3605.

15. Kitaura, K.; Ikeo, E.; Asada, T.; Nakano, T.; Uebayasi, M.*Chemical Physics Letters* **1999** , *313(3)* , 701-706.

16. Fukui, K. *Accounts of chemical research* **1981** ,*14(12)* , 363-368.

17. Schlegel, H. B. *Journal of Computational Chemistry***1982** , *3(2)* , 214-218.

18. Maeda, S.; Harabuchi, Y.; Ono, Y.; Taketsugu, T.; Morokuma, K.*International Journal of Quantum Chemistry* **2015** ,*115(5)* , 258-269.

19. Humphrey, W.; Dalke, A.; Schulten, K. *Journal of molecular graphics* **1996** , *14(1)* , 33-38.

20. Dennington, R.; Keith, T.; Millam, J. GaussView, version 5, 2009.

21. BIOVIA, D. S. Materials Studio, 2017.

22. Rion Dooley; Kent Milfeld; Chona Guiang; Sudhakar Pamidighantam; Allen, G. *Journal of Grid Computing* **2006** , *4(2)* , 195-208.

23. Srinivasarao, M.; Low, P. S. *Chemical Reviews* **2017** ,*117(19)* , 12133–12164.

24. Robert Zaleśny; Papadopoulos, M. G.; Mezey, P. G.; Leszczynski, J. Linear-Scaling Techniques in Computational Chemistry and Physics: Methods and Applications; Springer, 2011.

25. White, C. A.; Head-Gordon, M. *The Journal of chemical physics***1994** , *101(8)* , 6593-6605.

26. Challacombe, M.; Schwegler, E.; Almlöf, J. *The Journal of chemical physics* **1995** , *104(12)* , 4685-4698.

27. White, C. A.; Head-Gordon, M. *The Journal of chemical physics***1996** , *105(12)* , 5061-5067.

28. Challacombe, M.; Schwegler, E. *The Journal of chemical physics* **1997** , *106(13)* , 5526-5535.

29. Ochsenfeld, C.; White, C. A.; Head-Gordon, M. *The Journal of chemical physics* **1998** , *109(5)* , 1663-1669.

30. Ochsenfeld, C. *Chemical Physics Letters* **2000** ,*327* , 216–223.

31. He, X.; Zhang, J. Z. H. *The Journal of chemical physics***2005** , *122(3)* , 31103.

32. Li, W.; Li, S.; Jiang, Y. *The Journal of Physical Chemistry A***2007** , *111(11)* , 2193-2199.

33. Xie, W.; Orozco, M.; Truhlar, D. G.; Gao, J. *Journal of Chemical Theory and Computation* **2009** , *5(3)* , 459-467.

34. Gao, J.; Cembran, A.; Mo, Y. *Journal of Chemical Theory and Computation* **2010** , *6(8)* , 2402-2410.

35. Steinmann, C.; Ibsen, M. W.; Hansen, A. S.; Jensen, J. H. *PLoS One* **2012** , *7(9)* , e44480.

36. Wang, Y.; P.Sosa, C.; Cembran, A.; Truhlar, D. G.; Gao, J.*Journal of Physical Chemistry B,* **2012** , *116(23)* , 6781-6788.

37. He, X.; Zhu, T.; Wang, X.; Liu, J.; Zhang, J. Z. H. *Acc ChemRes* **2014** , *47(9)* , 2748-2757.

38. Jin, X.; Zhang, J. Z. H.; He, X. *Journal of Physical Chemistry A* **2017** , *121(12)* , 2053-2514.

39. Haili Xiao; Wang, X. *CHINA SCIENCE AND TECHNOLOGY ACHIEVEMENTS, in Chinese* **2018** *(4)* , 10.

40. Tokiwa, T.; Nakano, S.; Yamamoto, Y.; Ishikawa, T.; Ito, S.; Sladek, V.; Fukuzawa, K.; Mochizuki, Y.; Tokiwa, H.; Misaizu, F.; Shigeta, Y.*Journal of Chemical Information and Modeling* **2019** ,*59* , 25-30.

41. Yang, W. *Physical Review Letters* **1991** ,*66(11)* , 1438.

42. Gao, J. *The Journal of Physical Chemistry B* **1997** ,*101(4)* , 657-663.

43. Babu, K.; Gadre, S. R. *Journal of computational chemistry***2003** , *24(4)* , 484-495.

44. Liu, W.; Wang, F.; Li, L. *Journal of Theoretical Computational Chemistry* **2003** , *02(02)* , 257-272.

45. Xie, W.; Gao, J. *Journal of Chemical Theory and Computation***2007** , *3(6)* , 1890-1900.

46. Suarez, E.; Diaz, N.; Suarez, D. *Journal of chemical theory computational Biology and Chemistry* **2009** , *5(6)* , 1667-1679.

47. Ma, Y.; Liu, Y.; Ma, H. *The Journal of chemical physics***2012** , *136* , 024113.

48. Ma, Y.; Ma, H. *The Journal of Physical Chemistry A***2013** , *117* , 3655-3665.

49. Heifetz, A.; Trani, G.; Aldeghi, M.; MacKinnon, H. C.; McEwan, A. P.; Brookfield, A. F.; Chudyk, I. E.; Bodkin, M.; Pei, Z.; Burch, D. J.*Journal of medicinal chemistry* **2016** , *59(9)* , 4352-4363.

50. Heifetz, A.; Aldeghi, M.; Chudyk, E. I.; Fedorov, D. G.; Bodkin, M.; Biggin, P. C. *Biochemical Society Transactions* **2016** ,*44(2)* , 574-581.

51. Liu, J.; Zhang, J. Z. H.; He, X. *Physical Chemistry Chemical Physics* **2016** , *18(3)* , 1864-1875.

52. Singh, G.; Nandi, K. A.; Gadre, R. S. *Journal of Chemical Physics* **2016** , *144(10)* , 104102.

53. He, X.; Zhang, J. Z. H. *Journal of Chemical Physics***2006** , *124(18)* , 184703.

54. Wang, X.; Liu, J.; Zhang, J. Z.; He, X. *Journal of Physical Chemistry A* **2013** , *117(32)* , 7149-7161.

55. Fedorov, D.; Kitaura, K. In The Fragment Molecular Orbital Method:Practical Applications to Large Molecular Systems; Fedorov, D.; Kitaura, K., Eds.; CRC Press, 2009, p 5-36.

56. Nagata, T.; Fedorov, D. G.; Kitaura, K. In Linear-Scaling Techniques in Computational Chemistry and Physics; Zalesny, R.; Papadopoulos, M. G.; Mezey, P. G.; Leszczynski, J., Eds., 2011, p 17-64.

57. Fedorov, D. G. *Comput Mol Sci* **2017** , e1322.

58. Fedorov, D. G.; Nagataa, T.; Kitauraab, K. *Phys Chem Chem Phys* **2012** , *14(21)* , 7562-7577.

59. Tanaka, S.; Mochizuki, Y.; Komeiji, Y.; Okiyama, Y.; Fukuzawa, K.*Phys Chem Chem Phys* **2014** , *16* , 10310-10344

60. Fedorov, D. G.; Kitaura, K. *J Phys Chem A* **2016** ,*120(14)* , 2218-2231.

61. Steinmann, C.; Fedorov, D. G.; Jensen, J. H. *J Phys Chem A***2010** , *114* , 8705–8712.

62. Li, H.; Federov, D. G.; Nagata, T.; Kitaura, K.; Jensen, J. H.*Journal of Computational Chemistry* **2009** , *31(4)* , 778-790.

63. Vuong, V. Q.; Nishimoto, Y.; Fedorov, D. G.; Sumpter, B. G.; Niehaus, T. A. *J Chem Theory Comput* **2019** , *15* , 3008-3020.

64. Nakano, T.; Kaminuma, T.; Sato, T.; Akiyama, Y.; Uebayasi, M.; Kitaura, K. *Chemical Physics Letters* **2000** ,*318(6)* , 614-618.

65. Komeiji, Y.; Inadomi, Y.; Nakano, T. *Computational Biology and Chemistry* **2004** , *28(2)* , 155-161.

66. Gordon, S. M.; Michael, W. S. In Theory and Applications of Computatioanl Chemistry:The First Forty Years, 2005, p 1167-1189.

67. Inadomi, Y.; Takami, T.; Maki, J.; Kobayashi, T.; Aoyagi, M. In Parallel Computing: From Multicores and GPU's to Petascale, 2010, p 220-227.

68. Mazack, M. J. M.*http://mazack.org/cgi-bin/xpol.pl***2013** *(Access at Sep.30,2019)* .

69. Suenaga, M. *Journal of Computer Chemistry, Japan***2005** , *4(1)* , 25-32.

70. Qiu, Y.; Suo, B.; Zhang, B.; Jin, Z. *e-Science Technology & Application (China)* **2016** *(1)* , 15-23.

71. Chen, J.; Duan, Q.; JIn, Z.; Liu, Q.; Zhang, B.; Chi, X.*Application Research of Computers,China* **2012** ,*29(2)* , 431-437.

72. Waldo, J. *IEEE concurrency* **1998** , *6(3)* , 5-7.

73. M. J. Frisch, G. W. T., H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, Gaussian, Inc., Wallingford CT. **2016** .

74. Neidigh, W. J.; Fesinmeyer, R. M.; Andersen, H. N. H. *Nature Structural Biology* **2002** , *9(6)* , 425-430.

75. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A.*Journal of Computational Chemistry* **2004** , *25(9)* , 1157-1174.

76. Wang, J.; b, W. W.; Kollman, P. A.; Case, D. A. *Journal of Molecular Graphics and Modelling* **2006** , *25* , 247–260.

77. D.A. Case, I. Y. B.-S., S.R. Brozell, D.S. Cerutti, T.E. Cheatham, III, V.W.D. Cruzeiro, T.A. Darden, R.E. Duke, D. Ghoreishi, M.K. Gilson, H. Gohlke, A.W. Goetz, D. Greene, R Harris, N. Homeyer, S. Izadi, A. Kovalenko, T. Kurtzman, T.S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D.J. Mermelstein,

K.M. Merz, Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D.R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C.L. Simmerling, J. Smith, R. Salomon-Ferrer, J. Swails, R.C. Walker, J. Wang, H. Wei, R.M. Wolf, X. Wu, L. Xiao, D.M. York and P.A. Kollman. *University of California, San Francisco* **2016** .

78. Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, K. M.; Greenwood, J. *Journal of the American Chemical Society* **2015** , *137(7)* , 2695-2703.

79. Head-Gordon, M.; Pople, J. A.; Frisch, M. J. *Chemical Physics Letters* **1988** , *153(6)* , 503-506.

80. Adamo, C.; Barone, V. *The Journal of chemical physics***1999** , *110(13)* , 6158-6170.

81. Tao, J.; Perdew, P. J.; Staroverov, N. V.; Scuseria, E. G.*Physical Review Letters* **2003** , *91(14)* , 146401.

82. Yanai, T.; Tew, P. D.; Handy, C. N. *Chemical Physics Letters***2004** , *393(1-3)* , 51-57.

83. Zhao, Y.; Truhlar, G. D. *Theoretical Chemistry Accounts***2008** , *120(1-3)* , 215-241.

84. Sun, T.; Wang, Y. *Acta Physico-Chimica Sinica (China)***2011** , *27(11)* , 2553-2558.

85. Yu, H. S.; He, X.; Li, S. L.; Truhlar, D. G. *Chemical science***2016** , *7(8)* , 5032-5051.

86. Stefan, G.; Jens, A.; Stephan, E.; Helge, K. *Journal of Chemical Physics* **2010** , *132(15)* , 154104.

87. Smith, D. G.; Burns, L. A.; Patkowski, K.; Sherrill, C. D.*Journal of Physical Chemistry Letters* **2016** ,*7(12)* , 2197.

88. Goerigk, L.; Hansen, A.; Bauer, C.; Ehrlich, S.; Najibi, A.; Grimme, S. *Physical Chemistry Chemical Physics* **2017** ,*19(48)* , 32184.

89. Erez, R.; Segal, E.; Miller, K.; Satchi-Fainaro, R.; Shabat, D.*Bioorganic & Medicinal Chemistry* **2009** , *17(13)* , 4327–4335.

90. Leamon, C. P.; Reddy, J. A.; Klein, P. J.; Vlahov, I. R.; Dorton, R.; Bloomfield, A.; Nelson, M.; Westrick, E.; Parker, N.; Bruna, K.; Vetzel, M.; Gehrke, M.; Messmann, J. S. N. R. A.; LoRusso, P. M.; Sausville, E. A. *Journal of Pharmacology and Experimental Therapeutics* **2010** , *336(2)* , 326-343.

91. Sharm, A.; Kim, E.-J.; Shi, H.; Lee, J. Y.; Chung, B. G.; Kim, J. S.*Biomaterials* **2018** , *155* , 145–151.

92. Te Velde, G. t.; Bickelhaupt, F. M.; Baerends, E. J.; Fonseca Guerra, C.; van Gisbergen, S. J.; Snijders, J. G.; Ziegler, T.*Journal of Computational Chemistry* **2001** , *22(9)* , 931-967.

93. Plimpton, S.; Thompson, A.; Crozier, P.; Kohlmeyer, A.*http://lammps.sandia.gov (Access at Sep 5, 2019)* .

94. Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, E. A.; Berendsen, J. H. *Journal of computational chemistry***2005** , *26(16)* , 1701-1718.

95. Valiev, M.; Bylaska, J. E.; Govind, N.; Kowalski, K.; Straatsma, P. T.; Van Dam, J. H.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, L. T.*Computer Physics Communications* **2010** , *181(9)* , 1477-1489.

96. Werner, H. J.; Knowles , J. P.; Knizia, G.; Manby, R. F.; Schutz, M.*Wiley Interdisciplinary Reviews: Computational Molecular Science***2012** , *2(2)* , 242-253