CHARMM-GUI 10 Years for Biomolecular Modeling and Simulation

Sunhwan Jo,[a] Xi Cheng,[b] Jumin Lee,[c] Seonghoon Kim,[c] Sang-Jun Park,[c] Dhilon S. Patel,[c] Andrew H. Beaven,[d] Kyu Il Lee,[c] Huan Rui,[e] Soohyung Park,[c] Hui Sun Lee,[c] Benoît Roux,[e] Alexander D. MacKerell, Jr,[f] Jeffrey B. Klauda,[g] Yifei Qi,[c] and Wonpil Im*[c]

CHARMM-GUI, http://www.charmm-gui.org, is a web-based graphical user interface that prepares complex biomolecular systems for molecular simulations. CHARMM-GUI creates input files for a number of programs including CHARMM, NAMD, GROMACS, AMBER, GENESIS, LAMMPS, Desmond, OpenMM, and CHARMM/OpenMM. Since its original development in 2006, CHARMM-GUI has been widely adopted for various purposes and now contains a number of different modules designed to set up a broad range of simulations: (1) *PDB Reader & Manipulator*, *Glycan Reader*, and *Ligand Reader & Modeler* for reading and modifying molecules; (2) *Quick MD Simulator*, *Membrane Builder*, *Nanodisc Builder*, *HMMM Builder*, *Monolayer Builder*, *Micelle Builder*, and *Hex Phase Builder* for building all-atom simulation systems in various environments; (3) *PACE CG Builder* and *Martini Maker* for building coarse-grained simulation systems; (4) *DEER Facilitator* and *MDFF/xMDFF Utilizer* for experimentally guided simulations; (5) *Implicit Solvent Modeler*, *PBEQ-Solver*, and *GCMC/BD Ion Simulator* for implicit solvent related calculations; (6) *Ligand Binder* for ligand solvation and binding free energy simulations; and (7) *Drude Prepper* for preparation of simulations with the CHARMM Drude polarizable force field. Recently, new modules have been integrated into CHARMM-GUI, such as *Glycolipid Modeler* for generation of various glycolipid structures, and *LPS Modeler* for generation of lipopolysaccharide structures from various Gram-negative bacteria. These new features together with existing modules are expected to facilitate advanced molecular modeling and simulation thereby leading to an improved understanding of the structure and dynamics of complex biomolecular systems. Here, we briefly review these capabilities and discuss potential future directions in the CHARMM-GUI development project. © 2016 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.24660

Introduction

Modeling and simulation of biologically important molecules have come a long way since their inception.[1] Various computational methodologies have been developed not only to elucidate the mysteries in biological phenomena,[2–6] but also to predict the outcome when the systems or the conditions are changed. Nowadays, computations can be used to design new experiments to test the developed hypotheses.[7,8] Combined with the tremendous increase in computing power, biomolecular modeling and simulation have become routine tools used in many research laboratories.[9]

An issue that is not always fully appreciated is the difficulty to prepare and initiate the complex computational tasks needed to achieve a specific objective. Except for simple modeling and simulation tasks, the execution of most computational methods requires far more than a single "click." Advanced biomolecular modeling programs rely on sophisticated scripts, and thus handling all the relevant information to prepare a complex simulation system correctly often incurs a considerable amount of human time. This is an important issue, for beginners and experts alike, that cannot be overstated. Various software tools have been produced to address this need and offer support to the users.[10,11] Among these tools, the emergence of web-based modeling tools is notable. For example, CHARMMing,[12]

[a] *S. Jo*
Leadership Computing Facility, Argonne National Laboratory, 9700 Cass Ave, Argonne, Illinois
[b] *X. Cheng*
Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica Chinese Academy of Sciences, 555 Zuchongzhi Road, Pudong, Shanghai, 201203, China
[c] *J. Lee, S. Kim, S.-J. Park, D. S. Patel, K. I. Lee, S. Park, H. S. Lee, Y. Qi, W. Im*
Department of Biological Sciences and Bioengineering Program, Lehigh University, Pennsylvania
E-mail: wonpil@lehigh.edu
[d] *A. H. Beaven*
Department of Chemistry, The University of Kansas, Lawrence, Kansas
[e] *H. Rui, B. Roux*
Department of Biochemistry and Molecular Biology, The University of Chicago, Chicago, Illinois
[f] *A. D. MacKerell, Jr*
Department of Pharmaceutical Sciences School of Pharmacy, University of Maryland, Baltimore, Maryland
[g] *J. B. Klauda*
Department of Chemical and Biomolecular Engineering and the Biophysics Program, University of Maryland College Park, Maryland

Contract grant sponsor: NSF; Contract grant numbers: MCB-0918374, MCB-1157677, MCB-1516154, DBI-1145987, DBI-1707207, IIA-1359530 (to WI), MCB-1149187, DBI-1145652 (to JBK); Contract grant sponsor: NIH; Contract grant numbers: GM092950 (to WI), GM087519 (to WI and BR), GM072558 (to ADM and BR), GM070855, GM051501 (to ADM); Contract grant sponsor: Extreme Science and Engineering Discovery Environment (XSEDE) allocations, MCB-070009 (to WI) and MCB-100139 (to JBK); Contract grant number: NSF ACI-1053575.

Glycam Biomolecule Builder,[13] and MDWeb[14] provide graphical web interface for preparing chemical systems for molecular dynamics (MD) simulations. In addition, there is a battery of web-based tools for preparing custom force field (FF), such as ParamChem,[15,16] MATCH,[17] SwissParam,[18] GAAMP,[19] and RED.[20] Compared to downloadable software, web-based tools do not require program installation or upgrades on the user end and can be accessed using a web browser, rendering them "platform-independent" and easy to use.

Among various web-based modeling tools, CHARMM-GUI,[21] http://www.charmm-gui.org, sets out to simplify and generalize the protocols for building complex simulation systems and preparing simulation input files for widely used simulation packages, such as CHARMM,[22] NAMD,[23] GROMACS,[24] AMBER,[25] GENESIS,[26] LAMMPS,[27] Desmond,[28] OpenMM,[29] and CHARMM/OpenMM[30] to facilitate the usage of common and advanced simulation techniques. Molecular simulation systems are often comprised of multiple components, and in-depth knowledge of modeling software is often necessary to understand and build a sophisticated simulation system. In addition, generalizing the system building protocols is challenging because customization is often necessary. For example, building a realistic and physiologically relevant protein-membrane complex containing water, ions, protein, and various lipid types would require a considerable effort, even for an expert. CHARMM-GUI *Membrane Builder* successfully encapsulates such a sophisticated process into a well-defined workflow that can reproducibly generate a realistic protein-membrane complex or a membrane-only system within minutes to a few hours depending on the system size.[21,31–33]

Many online tools aim to provide a general modeling facility to users. One notable exception is the "What-If" server (http://swift.cmbi.ru.nl/whatif/) developed by Vriend et al..[34] The server provides a simple and easy-to-use interface to explore consequences of structural changes to a protein; for example, "what if a certain residue is mutated?" or "how would the solubility of the protein change?" By focusing on a task, instead of providing low-level functionality (e.g., introducing mutation), the server is able to simplify and streamline many aspects of interface design.

Inspired by such task-oriented design approach, our philosophy in developing CHARMM-GUI is less about providing the nuts and bolts of molecular modeling, but we instead focus on helping users to achieve a task, such as building a membrane system or solvating a protein, by providing a streamlined interface. This design principle helps our team to think of the workflow critically while designing the interface, which led CHARMM-GUI to be accessible to users with less experience in modeling tools and yet still useful to experts, especially for batch generation of systems. As with any design decision, there is a trade-off with the task-oriented approach that we have adopted in CHARMM-GUI. For example, a major limitation is that CHARMM-GUI contains many pre-determined

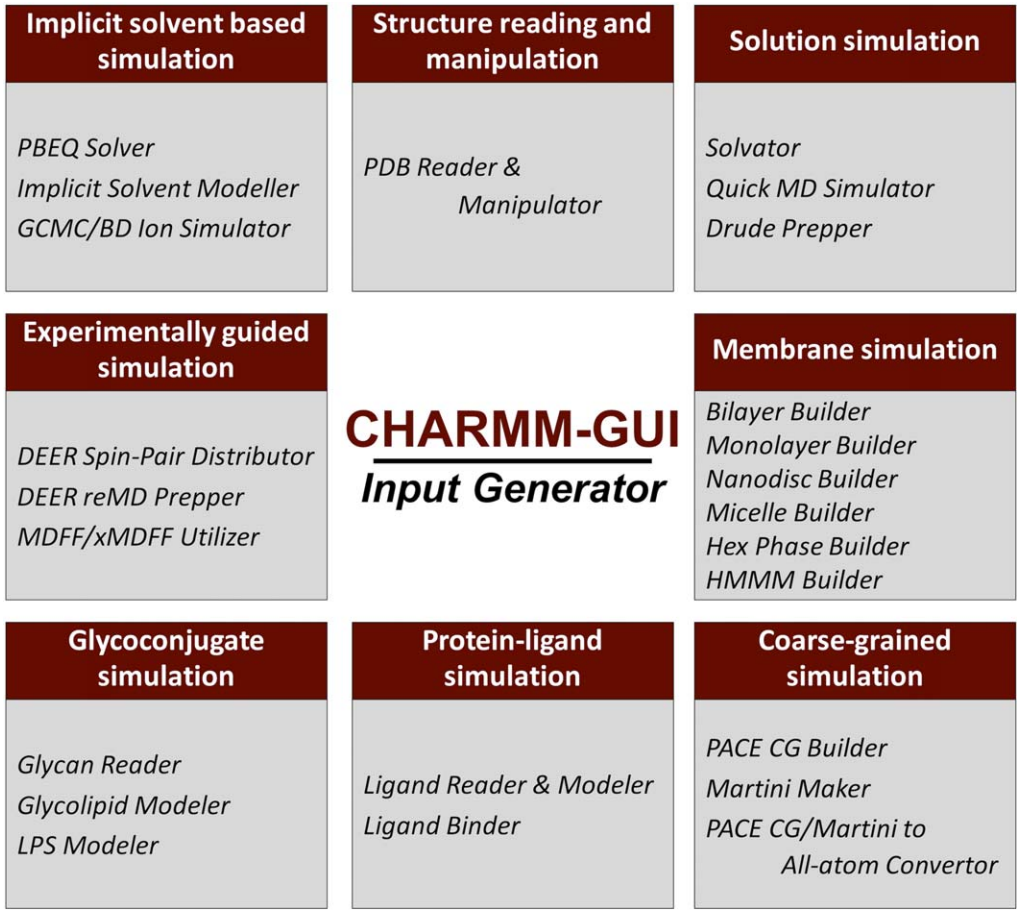

Figure 1. Schematic view of modules in CHARMM-GUI *Input Generator*. [Color figure can be viewed at wileyonlinelibrary.com]

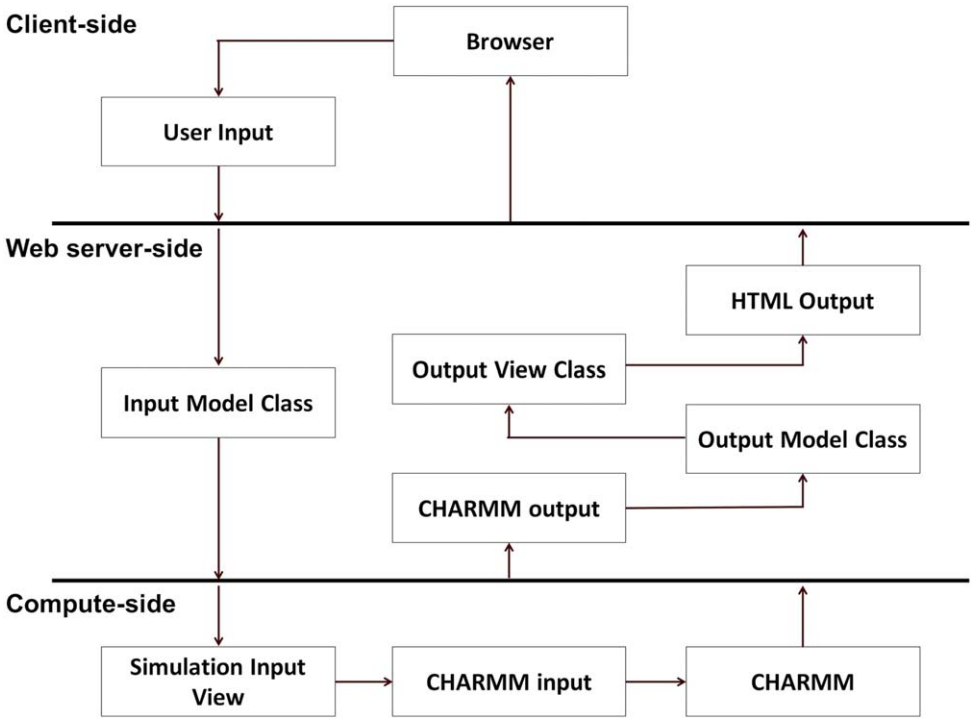

Figure 2. CHARMM-GUI model-view-controller (MVC) architecture pattern. [Color figure can be viewed at wileyonlinelibrary.com]

parameters (e.g., number of steps for minimization or the specifics of how to solvate a protein), which are hidden from users, potentially making CHARMM-GUI less appealing to more advanced users for their specific customization needs. To address such issues, CHARMM-GUI provides raw CHARMM input files, allowing users to optimize our default protocols or adopt more sophisticated approaches without having to create entire input files from scratch. This practice is particularly useful as the program CHARMM is now free for academic use (http://www.charmm.org).

Since its original development in 2006, CHARMM-GUI has been extended to a range of capabilities and now contains a number of different modules designed to set up a broad range of biological systems (Fig. 1). Many of the original CHARMM-GUI modules were developed as an in-house effort, but we have undertaken a number of collaborative efforts with other groups to develop new modules. Many novel computational methodologies require complex system setup and the use of sensible parameters. In our co-development, we focus on a limited number of specific use cases and implement a clear and concise user-interface encapsulating a complex workflow with sensible, tested parameters. In the remainder of this review, we briefly discuss different modules in CHARMM-GUI that are currently available and our ongoing efforts in bringing more complex features to users.

Functionalities of CHARMM-GUI *Input Generator*

Current *modus operandi*

CHARMM-GUI is designed as a web application for preparing a complex simulation system without the installation of any program or sophisticated knowledge in modeling software. At the same time, one of the requirements in the design phase of CHARMM-GUI was to provide a way to reuse/repurpose our methodologies for expert users. To achieve both goals, CHARMM-GUI is designed to generate CHARMM input scripts followed by executing CHARMM binary using the generated input. By doing so, users can download the generated CHARMM input scripts and reuse/repurpose them if needed.

The design of CHARMM-GUI closely follows the model-view-controller (MVC) architecture pattern, which is widely adopted in GUI design and software development. The conventional MVC pattern separates an application into three interconnected parts; a *model* directly manages data, logics, and rules of the application, a *view* renders information, and a *controller* accepts inputs from users and controls *model* or *view*. CHARMM-GUI uses an in-house MVC framework based on PHP programming language (Fig. 2). In CHARMM-GUI, there are two independent *model* and *view* pairs that are responsible for user interface generation and CHARMM script generation, respectively, and a single *controller* that converts user inputs and commands these *models*/*views*. Such modular design allows new modules to be developed rapidly without extensive knowledge of the other modules.

Although CHARMM is very flexible and has a scripting facility, there are certain cases that CHARMM cannot be directly used due to the nature of complex biomolecular system building, for example, converting CHARMM FF parameter to GROMACS, checking lipid tail penetration to cyclic groups, and so on. Therefore, CHARMM-GUI has adopted our own Python scripts and/or various other external programs to handle such complex cases. All the generated systems, the CHARMM input scripts, and Python scripts for data conversion are freely

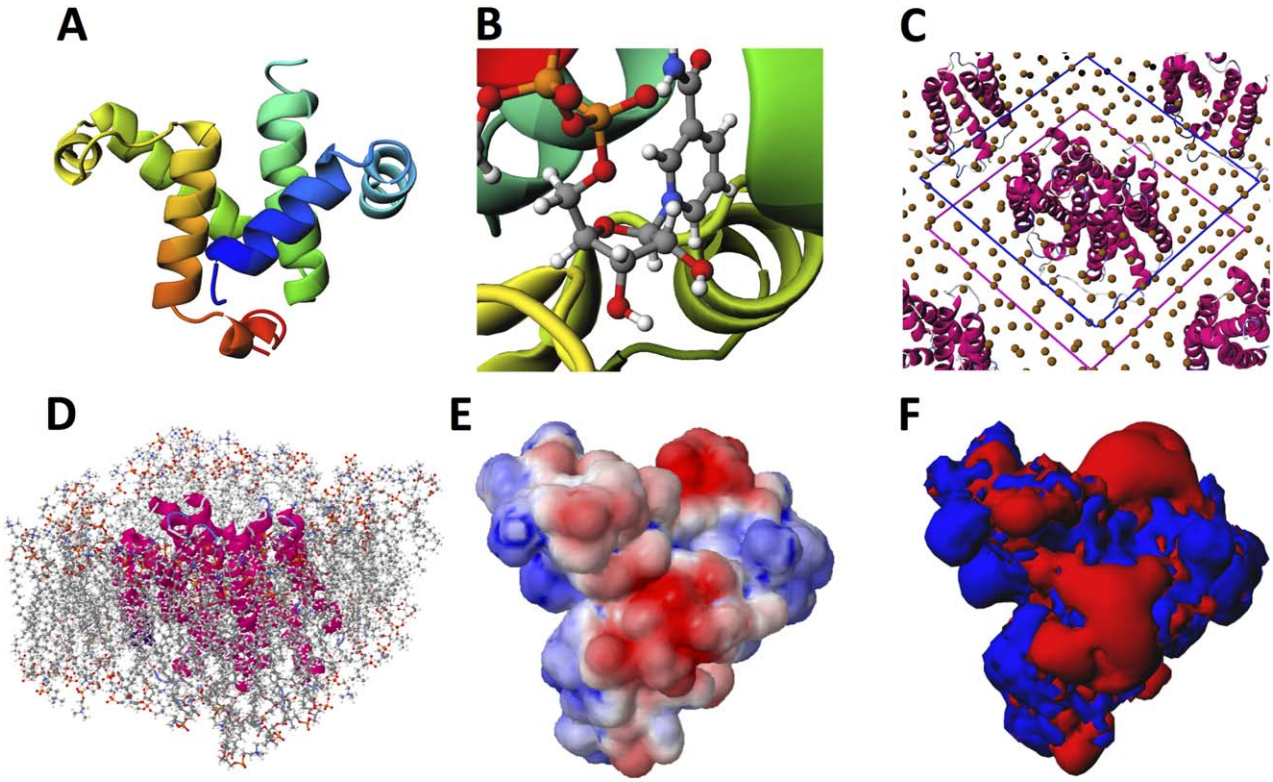

Figure 3. Molecular graphics views of A) KIX domain and co-activator KID (PDB:1KDX), B) DOX-P reductoisomerase with NADH (PDB:4ZQG), C) lipid-like pseudo atom packing around multidrug transporter EmrD (PDB:2GFP) with the primary system indicated by lines, and D) after replacement of lipid-like pseudo atoms in (C) with corresponding all-atom lipids (before equilibration), E) electrostatic potential on the solvent-accessible surface representation of KIX domain, and F) iso-electrostatic potential contour of KIX domain. [Color figure can be viewed at wileyonlinelibrary.com]

available to users by clicking the "download" button. Currently, CHARMM-GUI is made up of approximately 201,000 lines of code, including ~45,000 lines of PHP, ~25,000 lines of lipid library configuration files, ~17,000 lines of Python scripts, ~44,000 lines of HTML/CSS/JavaScript, and ~70,000 lines of CHARMM input templates.

PDB reading and manipulation

Reading a PDB file containing multiple components into a simulation program is not straightforward and is generally considered the first hurdle in a simulation project. It is typically even harder to introduce mutations, different protonation states of titratable residues, disulfide bonds, other post-translational modifications (such as phosphorylation, glycosylation, and lipid-tail linkers), naturally modified nucleic acids, or experimental modifications (such as adding spin labels, fluorophores, and unnatural amino acids). *PDB Reader & Manipulator* provides a flexible web interface to handle these and converts a PDB file (downloaded from RCSB,[35] http://www.rcsb.org, or uploaded by the user) into a protein structure file (PSF) that can be directly read by some simulation packages or readily converted to other formats, facilitating the setup and simulation of all-atom heterogeneous systems with the use of the comprehensive CHARMM additive FF that covers the full range of biomolecules.[36–40] In addition, since many PDB entries have missing protein residues, *PDB Reader & Manipulator* detects such missing residues and provides an option to model them using GalaxyFill.[41] If a user wants to only use some components of a PDB file (e.g., for solvating a small drug-like molecule in water or embedding a lipid molecule in a lipid bilayer), *PDB Reader & Manipulator* provides options to model the ligand(s) using CGenFF (CHARMM Generalized force field)[42,43] or ANTECHAMBER[44] if they do not exist already in the CHARMM additive FF.

RCSB has been transitioning from PDB to PDBx/mmCIF format. Using mmCIF is advantageous over the conventional PDB format that has fixed column widths, causing issues in the number of atoms, residue names, and/or atom names when an entry is longer than the corresponding width. In addition, mmCIF format is more portable because the data format specifications are included in the same file. *PDB Reader & Manipulator* is able to handle both PDB and PDBx/mmCIF formats. While *PDB Reader & Manipulator* exists as an independent module, it is the first step in all other modules.

Visualization

CHARMM-GUI uses JSmol[45] to visualize three-dimensional (3D) molecular structures, allowing users to check their system in each step. Currently, CHARMM-GUI provides a variety of viewing modes (Fig. 3): (1) protein view for protein-only structures with ribbon diagram rendering; (2) protein/membrane (or ligand) complex view with lipids as CPK and protein as

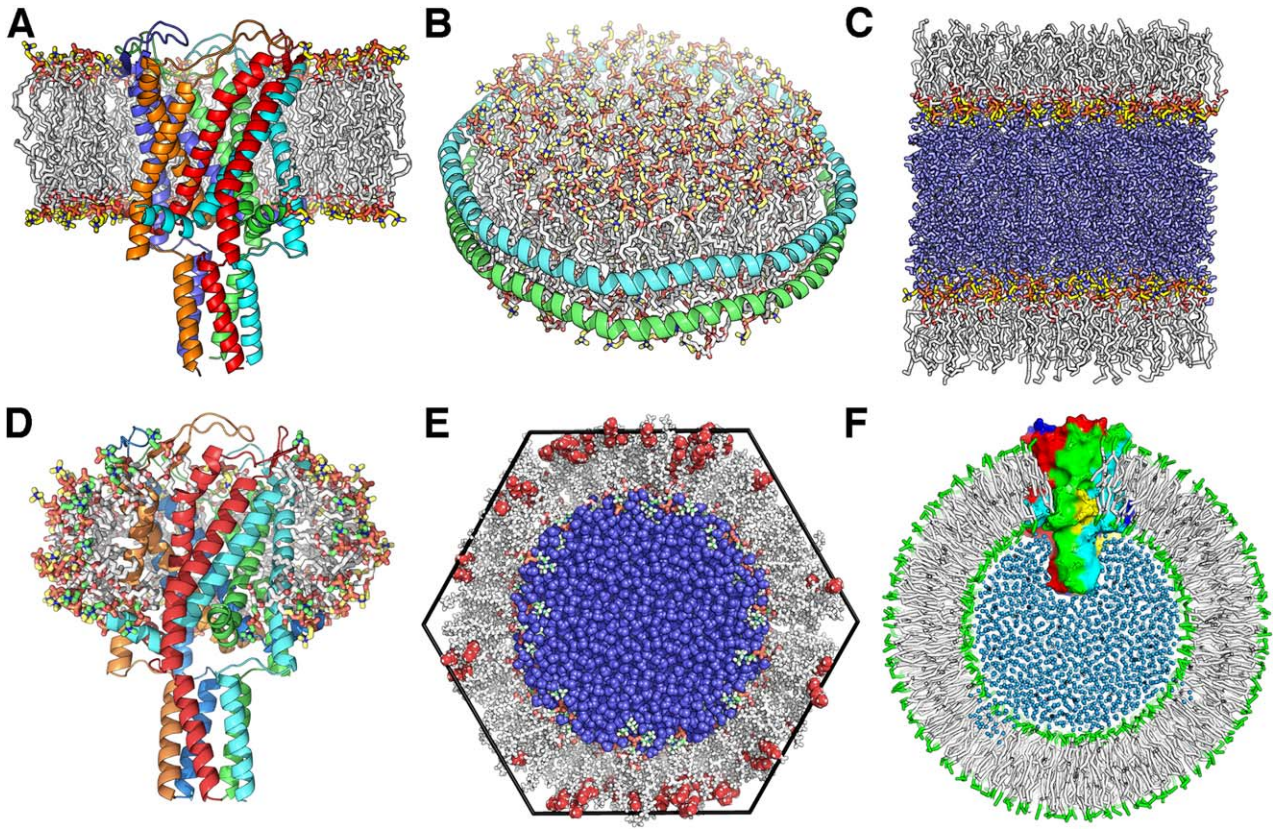

Figure 4. Molecular graphics views of A) bilayer, B) nanodisc, C) monolayer, D) micelle, E) inverse hexagonal lipid phase, and F) vesicle. The mechanosensitive channel of large conductance (MscL) is embedded in a bilayer, a micelle, and a vesicle. Water molecules are shown as blue spheres only in (C), (E), and (F). The alkane molecules in the hexagonal phase system are shown as red spheres. [Color figure can be viewed at wileyonlinelibrary.com]

ribbon diagram; (3) orientation view for displaying both upper and lower planes indicating the bilayer hydrophobic core; (4) packing image view for lipid-like pseudo atoms with spheres (for both primary and neighboring image systems); and (5) electrostatic potential view for displaying electrostatic potential or contour maps obtained from the *PBEQ Solver*. Users do not need to install any additional program to display/render the structures in CHARMM-GUI, and the concise workflow with an appropriate mode in each building step increases the viewing speed.

Solution simulation

Starting from *PDB Reader & Manipulator*, input files for standard MD (equilibration and production) simulations of globular proteins and/or other molecules in aqueous solvent environments can be quickly prepared using *Quick MD Simulator*. All the simulation input files are prepared to use periodic boundary conditions (PBC) with the particle-mesh Ewald (PME) method[46] for long-range electrostatic interactions. In general, all simulation protocols in CHARMM-GUI are optimized for each FF used in different modules, following the principles of the original FF developments. If the developmental protocol is not supported in a certain MD engine, a set of available parameters has been tested and examined to optimize the protocol. Reference 47 describes the optimized protocols of all-atom simulations using the CHARMM FF (for solution and other simulations).

Membrane simulation

Biological membranes are complex in terms of lipid types and their compositions,[48] and the lipid molecules may interact with proteins in a specific manner. To better represent a native environment, more fatty acid, lipid, and detergent types have been added to CHARMM-GUI. Although lipid bilayers are the most common in biological systems, other membrane environments such as monolayer, micelle, nanodisc, hexagonal lipid phase, and vesicle are commonly used in experiments and thus require computational tools to build them (Fig. 4).

Membrane Builder[31–33] helps users to generate a protein-membrane complex with homogeneous or heterogeneous bilayers, as well as membrane-only bilayers. As of 2016, *Membrane Builder* has 295 lipid types in the context of CHARMM additive FF including phosphoinositides, cardiolipin, sphingolipids, bacterial lipids, sterols, and fatty acids, which makes it possible to build a realistic biological membrane system. Since RCSB PDB structures do not contain the orientation information of a membrane protein relative to lipid bilayers (whose normal is pre-defined to be parallel to the Z-axis and center is located at $Z = 0$), users can use the OPM database (http://opm.phar.umich.edu)[49] to obtain pre-oriented protein structures.

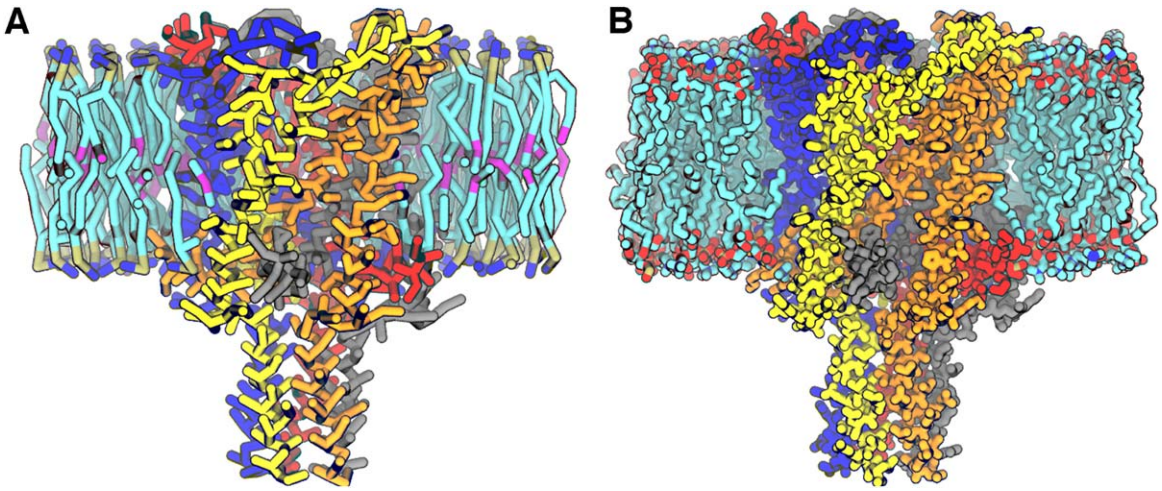

Figure 5. Conversion of A) a Martini bilayer system to B) an all-atom system. The mechanosensitive channel MscL protein is embedded in a POPC bilayer. Water molecules and ions are not shown for clarity. Different subunits of the MscL are drawn with different colors. [Color figure can be viewed at wileyonlinelibrary.com]

Membrane Builder also provides several options to orient proteins. We have optimized the automated building process including system size determination as well as generation of lipid bilayer, water, and ions. Any complex (homogeneous or heterogeneous) bilayer system can be generated by the so-called "replacement method"[50,51] that first packs the lipid-like pseudo atoms (Fig. 3C), and then replaces them with lipid molecules one at a time by randomly selecting a lipid molecule from a lipid structural library.[3,32] Using the replacement method, it generates nicely packed lipid molecules around a protein, although *Membrane Builder* provides an insertion method for limited homogeneous bilayer system building. Reference 31 describes both the replacement and insertion methods in detail. For proteins with water-filled lumens, *Membrane Builder* estimates the pore's solvent accessible area profile and hydrates the lumen. Careful checks of lipid membrane building are included to avoid unphysical structures, such as penetration of acyl chains into ring structures in sterols, aromatic residues, and carbohydrates.[33] The optimized equilibration scripts prevent any unwanted structural changes in lipids and embedded proteins, so that production simulations can probe physically realistic behavior of biomolecules.

Nanodisc Builder can be used to build a nanodisc system (with/without protein) in which two membrane scaffolding proteins (MSPs) wrap around a lipid bilayer to form a discoidal structure.[52,53] Nanodiscs have been widely used in biophysical and biochemical studies of membrane proteins. As there is no structural information of the MSPs in nanodiscs, *Nanodisc Builder* uses pre-assembled MSP structures based on the double-belt model[54] by stacking two MSP monomers against each other. Nine different MSPs that form different sizes of nanodiscs are available. In addition to providing all-atom nanodisc simulation inputs, *Nanodisc Builder* is also available in *PACE CG Builder*[55] and *Martini Maker*[56] for coarse-grained (CG) simulations.

Monolayer Builder helps users to generate a series of CHARMM input files necessary to build a protein-monolayer (or monolayer-only) system. The building procedure of a monolayer system is the same as that of a bilayer system except that a bilayer system is converted to a monolayer system by manipulations of the PBC after assembling all the system components.

Micelle Builder[57] helps users to build homogeneous/heterogeneous micelle and protein-micelle systems. Currently, *Micelle Builder* provides 20 detergent types. Considering the size and shape of the transmembrane segments of the protein, *Micelle Builder* arranges detergents in a micelle of a torus shape. Therefore, protein-micelle complexes can be built for proteins with any shape.[58]

Hex Phase Builder has been developed to probe a lipid phase that exists at high temperature or low water concentration regime (for a review, see Ref. [59]) and has been used in biophysical experiments and simulations to gather information on lipids at high curvature.[60–65] *Hex Phase Builder* uses a building procedure similar to *Membrane Builder* except that lipids are placed around a water cylinder instead of on a plane.

HMMM Builder[66] aids in studies that are hindered by the slow diffusion of lipids that exists in conventional all-atom membrane simulations, making it difficult to sample large rearrangement in a relatively short simulation time. *HMMM Builder* provides users with input files for simulating membrane systems (with/without proteins) using the highly mobile membrane-mimetic (HMMM) model,[67] in which the lipid tails are replaced with small organic molecules (currently 1,1-dichloroethane, i.e., DCLE), resulting in acceleration of lipid diffusion by 1–2 orders of magnitude. *HMMM Builder* supports all lipid types available in *Membrane Builder*, and the user can choose the position to trim an acyl chain and lipid packing ratio. In addition to building HMMM systems, *HMMM Builder* also provides a conversion tool between HMMM and full-length lipid systems, allowing users to easily take advantage of both HMMM and full-length membranes.

Coarse-grained simulation

Coarse-graining is a popular approach to model large simulation systems. *Martini Maker*[56] and *PACE CG Builder*[55] provide

simulation system preparations using the Martini FF[68] and the PACE FF,[69] respectively. The Martini model is widely adopted to represent lipid molecules with on average four heavy atoms mapped to one CG atom. *Martini Maker* facilitates Martini system building for solution, micelle, nanodisc, bilayer, and vesicle simulations as well as simulations starting with randomly distributed detergent/lipid molecules. The supported versions of the Martini FF include standard Martini,[68,70,71] Martini with polarizable water,[72,73] Dry Martini,[74] and ElNeDyn using elastic networks for proteins.[75] The generated simulation input files are ready for use with the latest GROMACS simulation software in which Martini simulations have been optimized. The Martini model is suitable for CG lipid molecules, but proteins are not often suitable to model in the same manner. The PACE model has been developed to perform CG simulations of united-atom (UA) protein with the Martini water and lipid FF.[68] Compared to the corresponding all-atom simulations using a 2-fs time step, the PACE simulation reduces the number of atoms by a factor of 10, and allows an integration time step of 5 fs, which speeds up the simulation about 30 times. The PACE FF provides more details for the protein than the Martini FF, and could be a superior feature in applications for modeling/refining large, complicated biological systems.

After performing CG simulations, it is sometimes desired to convert the CG system back to an all-atom representation. However, CG to all-atom mapping is a difficult problem. In this context, *Martini Maker* and *PACE CG Builder* provide back conversion of a CG simulation system to an all-atom system using a simple but effective method (Fig. 5). The Martini protein is converted using the backward.py script[76] and the PACE protein is converted by adding the hydrogen atoms back using CHARMM HBUILD functionality. Different from other approaches,[76–78] the CG lipids are converted into all-atom lipids using a lipid structure library (used in *Membrane Builder*) by aligning a randomly selected all-atom structure to the CG lipid. The CG waters are converted by mapping one CG water molecule back to four all-atom water molecules in a tetrahedron arrangement.

Protein–ligand simulation

As computer-aided drug design becomes increasingly popular, the number of MD simulations of drug-like molecules is on the rise.[79] Having a well-validated FF is extremely important to obtain meaningful results. *Ligand Reader & Modeler* helps both beginners and experts to prepare FF parameters conveniently by searching for a ligand in the entire CHARMM FF or using CGenFF. Users are able to upload a ligand to *Ligand Reader & Modeler* in various ways, and the uploaded structure can be interactively modified on the sketchpad (powered by Marvin JS, http://www.chemaxon.com). In addition, it is possible to draw chemical substitution sites and substituents in each site to generate combinatorial structures and corresponding FF parameters. Finally, the output can be used in other CHARMM-GUI modules to build a simulation system.

Advanced free energy perturbation molecular dynamics (FEP/MD) simulation methods[80–85] can be used to calculate absolute binding free energies of protein–ligand complexes. Typically, setting up a FEP/MD simulation is very difficult because these methods rely on various biasing energy restraints to enhance the convergence of the calculations. *Ligand Binder*[86] provides the standardized CHARMM input files for calculations of absolute binding free energies using FEP/MD simulations. A number of features, such as implicit treatment of bulk water,[87,88] sophisticated decomposition scheme,[89,90] or Hamiltonian-tempering replica-exchange simulation,[91,92] are implemented to conveniently set up the FEP/MD simulations in a highly customizable manner, thereby permitting an accelerated throughput of this important class of computations while decreasing the possibility of human errors.

Glycoconjugate simulation

Carbohydrate moieties, referred to as glycans, can be covalently attached to proteins (glycoproteins) and lipids (glycolipids), or exist as free ligands (Fig. 6). Understanding how glycosylation affects protein structure, dynamics, and function is an emerging and challenging problem in biology. Notably, about 30% of carbohydrate structures in the PDB contain at least one error regarding the sugar-type assignment.[93] As shown in Figure 7, *Glycan Reader* greatly simplifies the reading of PDB and PDBx/mmCIF structure files containing glycans using

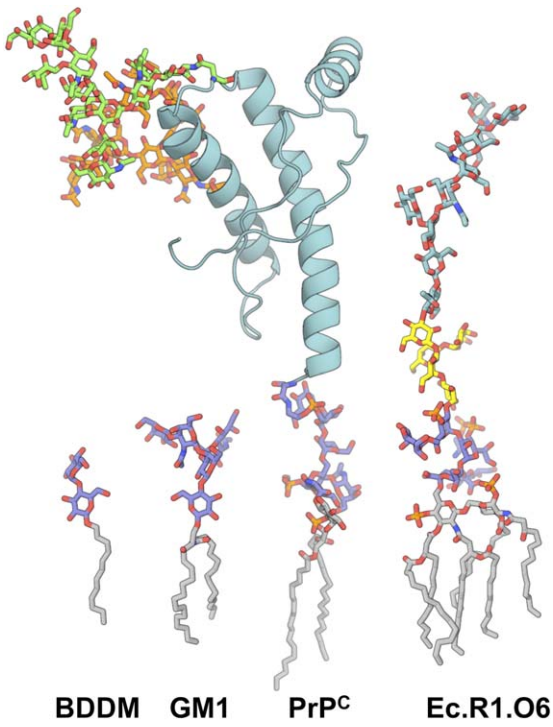

Figure 6. Structure of glycoconjugates: BDDM for dodecyl-β-D-maltoside, GM1 for monosialotetrahexosyl ganglioside, PrP^C for a glycosylphospatidylinositol (GPI)-anchored human prion protein (cyan) with two N-glycans (green),[147] and Ec.R1.O6 for *E. coli* lipopolysaccharide (LPS) with R1 core (blue for the inner core and yellow for the outer core) and two repeating units of O6 antigen (cyan).[148] [Color figure can be viewed at wileyonlinelibrary.com]

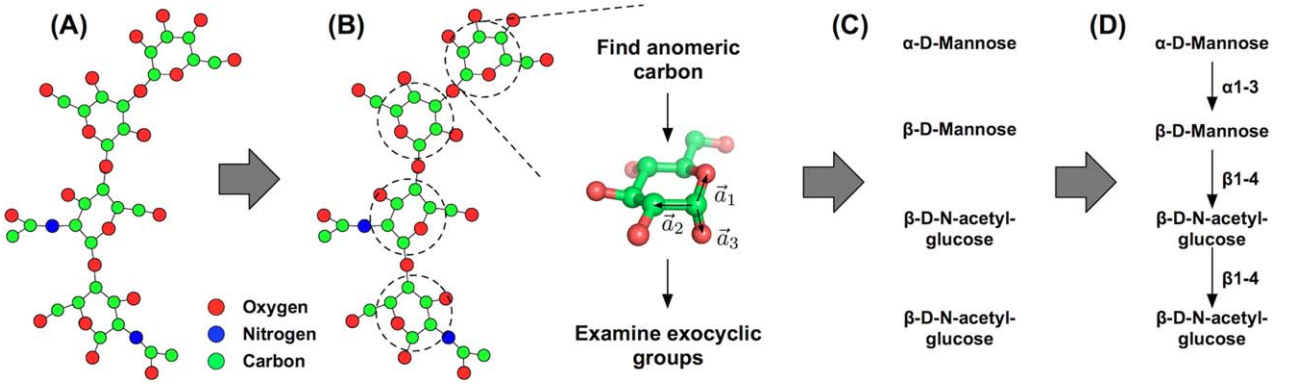

Figure 7. Illustration of carbohydrate annotation procedure in *Glycan Reader*. A) Molecular topology is built using HETATM and CONECT records in a PDB file. B) Potential carbohydrate molecules are examined for anomeric carbon, stereochemistry of each ring carbon atoms, and exocyclic groups. C) Carbohydrate type is annotated. D) Glycosidic linkages are assigned between monosaccharides. Figure reproduced with permission from *Journal of Computational Chemistry*.[93] [Color figure can be viewed at wileyonlinelibrary.com]

graph representations. Following considerable efforts, *Glycan Reader* is able to recognize all major carbohydrate chemical modifications in the PDB and generate their structure models. While it exists as a separate module, *Glycan Reader* is linked to other functional modules as part of *PDB Reader & Manipulator*, allowing users to easily generate molecular simulation systems with carbohydrate or glycoprotein, and visualize the electrostatic potential on glycoprotein surfaces.

Glycolipids have been widely studied because of their importance in a range of biological functions. However, computational studies of glycolipids are still challenging due to their structural complexity arising from glycosidic linkages between sugars. *Glycolipid Modeler* facilitates generation of an appropriate glycolipid structure. The module provides two options to generate a glycolipid structure and the corresponding PSF: (1) pre-defined glycolipid sequences and (2) glycolipid sequences specified by users. Currently, *Glycolipid Modeler* supports 171 pre-defined glycolipids with 51 lipid types and 11 carbohydrate types. In addition, it allows users to introduce various chemical modifications to carbohydrates.

LPS Modeler is designed to simplify generation of structural models of complex lipopolysaccharide (LPS) molecules. *LPS Modeler* provides the options to select bacterial type, lipid A type, core type, the number of O-antigen repeating units, and O-antigen type. Currently, *LPS Modeler* supports LPS structures from 14 bacteria species with 32 lipid A types, 49 core types, and 222 O-antigen types. The selected LPS sequence is displayed, so that users can check and modify the sugar sequence.

Experimentally guided simulation

Double electron–electron resonance (DEER) is a powerful technique to obtain distance information in the range of 18–80 Å by measuring the dipolar coupling between two unpaired electron spins. The distance distributions obtained from DEER experiments provide valuable protein structural information in a specific functional state. *DEER Spin-Pair Distributor* provides an interface to calculate the spin-pair distance distribution of labeled sites in a protein using MD simulations. Specially parameterized dummy spin-labels that mimic the methanethiosulfonate (MTS) spin labels are attached to user-specified labeling sites.[94,95] The simulation is performed for 10 ns using CHARMM on the server, which takes from 30 min (for a 60-residue protein) to 10 hours (for a 5000-residue protein). Following the link in the email sent to users after the job is done, the distributions between any labeled pair could be displayed on the web browser. The calculated distribution can be used to guide the selection of the label sites for further experiments as well as validate different protein structure models.

The distance distributions from DEER experiments can also be used with computational modeling and simulation to model/refine the protein structure that best match the DEER data. Restrained-ensemble MD (reMD) has been recently developed to simultaneously incorporate multiple distance distribution constraints for MD simulation.[94,96,97] In this method, each spin label is represented using multiple molecular fragments in all-

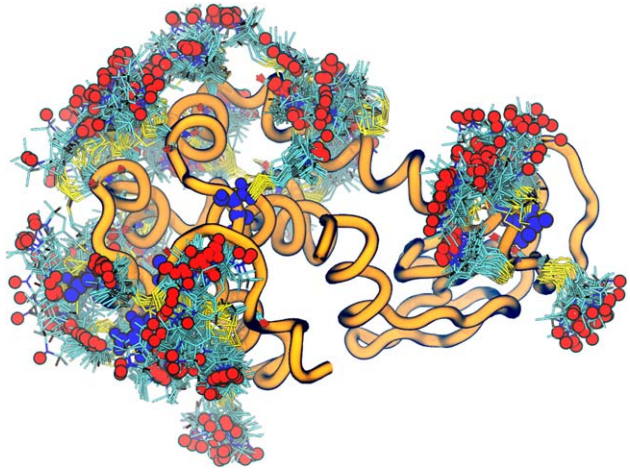

Figure 8. reMD simulation of lysozyme in solution. Multiple spin labels are added to one protein residue with the nitroxide groups shown as red spheres. The protein is shown as orange ribbon. The backbone atoms of the restrained residues are shown in blue spheres. Water molecules are removed for clarity. [Color figure can be viewed at wileyonlinelibrary.com]

atom detail (Fig. 8), and the distance distribution between two selected labels is restrained to the corresponding experimental distance distribution. Taking advantage of the spin label function in *PDB Reader & Manipulator,*[98] *DEER reMD Prepper* provides a quick and easy way to setup an reMD simulation.

MDFF and xMDFF are two structural refinement methods that incorporate experimental data from cryo-electron microscopy (cryo-EM) and X-ray crystallography into MD simulations.[99,100] The EM and X-ray data are added to the molecular mechanics FF as an external potential and the simulation is performed in real space. An important feature of these methods is that the macromolecule remains flexible and stereochemically correct during the simulation. *MDFF/xMDFF Utilizer* helps users to prepare MDFF and xMDFF simulation systems using the CHARMM all-atom or PACE CG FF in various environments including vacuum, implicit solvent, explicit solvent, and bilayers.

Implicit solvent based simulation

PBEQ Solver[101] allows the characterization of the electrostatic potential on a macromolecular surface by solving the Poisson–Boltzmann (PB) equation using the PBEQ module[102,103] in CHARMM. In addition, one can use this module to calculate (1) solvation free energy of biomolecules, (2) protein–protein (DNA or RNA) electrostatic interaction energy, and (3) pKa of a selected titratable residue in proteins.

Implicit solvent methods are a popular approximate description of solvent or membrane environments and CHARMM offers various implicit solvent models that users can apply to their systems. *Implicit Solvent Modeler* helps users to setup various implicit solvent models in CHARMM, and run molecular simulations with Langevin dynamics. The available implicit solvent models are: *ACE* (Analytical Continuum Electrostatics Potential),[104] *EEF1/IMM1* (Effective Energy Function),[105,106] *GBMV* (Generalized Born using Molecular Volume),[107,108] *GBSW* (Generalized Born with a simple Switching),[109,110] *SASA* (Solvent Accessible Surface Area implicit solvation model),[111] and *SCPISM* (Screened Coulomb Potentials Implicit Solvent Model).[112] In addition, users can also set up molecular simulations of membrane(-bound) proteins in membrane environments using EEF1/IMM1,[106] GBMV,[108] and GBSW.[109]

Brownian dynamics (BD) using a realistic, accurate potential of mean force (based on implicit solvent approximation) is an effective approach for simulating ion transport through wide aqueous molecular pores.[113–115] *GCMC/BD Ion Simulator*[116] prepares input files for grand canonical Monte Carlo (GCMC) BD simulations of channel proteins.[117,118] The webserver is designed to help users to avoid most of the technical difficulties and issues encountered in setting up and simulating complex pore systems.

Supported force fields and MD engines

Depending on the purpose of the simulation and the characteristics of the simulation system, different simulation methods and FFs can be applied. For example, to investigate atomistic structure and dynamic properties, an all-atom simulation system and a specific all-atom FF are necessary, while a large biological system may require CG methods. CHARMM-GUI supports various FFs: the CHARMM FF for all-atom additive simulations,[36–40] the Martini[68] and PACE CG FFs[69] for CG simulations, and the Drude FF[119,120] for polarizable all-atom simulations. Note that a Drude FF-based simulation system can be conveniently set up through *Drude Prepper* by simply uploading a PSF file and an equilibrated coordinate file obtained from MD simulation using the non-polarizable all-atom CHARMM FF.

CHARMM-GUI has been supporting various MD engines such as CHARMM,[22] NAMD,[23] GROMACS,[24] AMBER,[25] OpenMM,[29] and CHARMM/OpenMM,[30] and was recently extended to support GENESIS,[26] LAMMPS,[27] and Desmond.[28] While CHARMM, NAMD, OpenMM, CHARMM/OpenMM, and GENESIS support the standard CHARMM topology and parameter file formats (e.g., psf, prm, rtf, and str), other MD engines use different topology and parameter formats, requiring format conversion. We have developed several Python programs to generate the corresponding topology and parameter files (top and itp for GROMACS, dms for Desmond, and data file for LAMMPS) using CHARMM topology and parameter files, so that users can readily use the simulation input and data files for any supported MD engine.

Concluding Discussion

We have briefly reviewed the current functional modules in CHARMM-GUI, demonstrating diverse biomolecular systems that CHARMM-GUI supports to carry out innovative and novel biomolecular modeling and simulation research to acquire insight into structures, dynamics, and underlying mechanisms of biomolecular systems. It should be stressed that CHARMM-GUI also improves the reproducibility and quality of biomolecular simulations by providing simulation input files with generally accepted standards and well-validated protocols for all major simulation programs. The CHARMM-GUI development project is ongoing. For example, CHARMM-GUI currently supports a system building up to 3 million atoms, but as larger spatial scales, longer time scales, and higher levels of realism are becoming possible and necessary, we plan to dynamically adjust this limit in the future. In addition, more modules will be available in *Input Generator* for advanced simulations techniques, such as QM and QM/MM calculations,[121–124] NMR structure calculation and refinement,[125–128] multisite λ-dynamics,[129–131] constant pH simulations,[132–143] grid-based energy correction map,[144,145] and various replica-exchange enhanced sampling methods.[146] In terms of system building, our effort will be placed toward making *Glycolipid Modeler* and *LPS Modeler* available in *Membrane Builder* and other modules, enabling crowding and cellular-scale system modeling, and extending CHARMM-GUI for nanoscale material sciences. The current and future developments with more tutorials and illustrations are expected to open a door allowing CHARMM-GUI users to perform even more advanced biomolecular and nanomaterial simulations of a diverse range of chemical systems.

Keywords: Coarse-grained simulation · Glycan · Membranes · mmCIF · Protein-ligand interactions

How to cite this article: S. Jo, X. Cheng, J. Lee, S. Kim, S.-J. Park, D. S. Patel, A. H. Beaven, K. I. Lee, H. Rui, S. Park, H. S. Lee, B. Roux, A. D. MacKerell, Jr, J. B. Klauda, Y. Qi, W. Im. *J. Comput. Chem.* **2017**, *38*, 1114–1124. DOI: 10.1002/jcc.24660

[1] M. Karplus, J. Kuriyan, *Proc Natl Acad Sci U S A* **2005**, *102*, 6679.
[2] B. Roux, R. MacKinnon, *Science (New York, N.Y.)* **1999**, *285*, 100.
[3] W. Im, B. Roux, *J Mol Biol* **2002**, *319*, 1177.
[4] W. Im, J. Lee, T. Kim, H. Rui. *J Comput Chem* **2009**, *30*, 1622.
[5] K. Lindorff-Larsen, et al., *Science (New York, N.Y.)* **2011**, *334*, 517.
[6] T. Mori, et al., *Biochim Biophys Acta (BBA) - Biomembr* **2016**, *1858(7,PartB)*, 1635.
[7] S. Bernèche, B. Roux, *Nature* **2001**, *414*, 73.
[8] N. Huang, N. K. Banavali, A. D. MacKerell, *Proc Natl Acad Sci* **2003**, *100*, 68.
[9] R. O. Dror, et al., *Annu Rev Biophys* **2012**, *41*, 429.
[10] B. Knapp, W. Schreiner, *Bioinform Biol Insights* **2009**, *3*, 103.
[11] Q. Cui, R. Nussinov, *PLOS Comput Biol* **2014**, *10*, e1003786.
[12] B. T. Miller, et al., *J Chem Inform Model* **2008**, *48*, 1920.
[13] *Glycam Biomolecule Builder.* Available from: http://www.glycam.com.
[14] A. Hospital, et al., *Bioinformatics* **2012**, *28*, 1278.
[15] K. Vanommeslaeghe, A. D. MacKerell, Jr, *J Chem Inform Model* **2012**, *52*, 3144.
[16] K. Vanommeslaeghe, E. P. Raman, A. D. MacKerell, Jr, *J Chem Inform Model* **2012**, *52*, 3155.
[17] J. D. Yesselman, et al., *J Comput Chemi* **2012**, *33*, 189.
[18] V. Zoete, et al., *J Comput Chem* **2011**, *32*, 2359.
[19] L. Huang, B. Roux, *J Chem Theory Comput* **2013**, *9*, 3543.
[20] E. Vanquelef, et al., *Nucleic Acids Res* **2011**, *39(suppl2)*, W511.
[21] S. Jo, et al., *J Comput Chem* **2008**, *29*, 1859.
[22] B. R. Brooks, et al., *J Comput Chem* **2009**, *30*, 1545.
[23] J. C. Phillips, et al., *J Comput Chem* **2005**, *26*, 1781.
[24] M. J. Abraham, et al., *SoftwareX* **2015**, *1–2*, 19.
[25] D. A. Case, et al., *J Comput Chem* **2005**, *26*, 1668.
[26] J. Jung, et al., *Wiley Interdiscip Rev Comput Mol Sci* **2015**, *5*, 310.
[27] S. Plimpton, *J Comput Phys* **1995**, *117*, 1.
[28] K. J. Bowers, et al., Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters, In Proceedings of the ACM/IEEE SC 2006 Conference, **2006**.
[29] P. Eastman, et al., *J Chem Theory Comput* **2013**, *9*, 461.
[30] E. J. Arthur, C. L. Brooks, *J Comput Chem* **2016**, *37*, 927.
[31] S. Jo, T. Kim, W. Im, *PLoS ONE* **2007**, *2*, e880.
[32] S. Jo, et al., *Biophys J* **2009**, *97*, 50.
[33] E. L. Wu, et al., *J Comput Chem* **2014**, *35*, 1997.
[34] G. Vriend, *J Mol Graph* **1990**, *8*, 52.
[35] H. M. Berman, et al., *Nucleic Acids Res* **2000**, *28*, 235.
[36] J. B. Klauda, et al., *J Phys Chem B* **2010**, *114*, 7830.
[37] E. J. Denning, et al., *J Comput Chem* **2011**, *32*, 1929.
[38] K. Hart, et al., *J Chem Theory Comput* **2011**, *8*, 348.
[39] O. Guvench, et al., *J Chem Theory Comput* **2011**, *7*, 3162.
[40] R. B. Best, et al., *J Chem Theory Comput* **2012**, *8*, 3257.
[41] E. A. Coutsias, et al., *J Comput Chem* **2004**, *25*, 510.
[42] K. Vanommeslaeghe, et al., *J Comput Chem* **2010**, *31*, 671.
[43] W. Yu, et al., *J Comput Chem* **2012**, *33*, 2451.
[44] Case, D.A., et al., *AMBER 14*. **2014**: University of California, San Francisco.
[45] *Jmol: an open-source Java viewer for chemical structures in 3D*. Available from: http://www.jmol.org.
[46] T. Darden, D. M. York, L. G. Pedersen, *J Chem Phys* **1993**, *98*, 10089.
[47] J. Lee, et al., *J Chem Theory Comput* **2016**, *12*, 405.
[48] G. van Meer, D. R. Voelker, G. W. Feigenson, *Nat Rev Mol Cell Biol* **2008**, *9*, 112.
[49] M. A. Lomize, et al., *Bioinformatics* **2006**, *22*, 623.
[50] T. B. Woolf, B. Roux, *Proc Natl Acad Sci* **1994**, *91*, 11631.
[51] T. B. Woolf, B. Roux, *Proteins: Struct Funct Bioinform* **1996**, *24*, 92.
[52] I. G. Denisov, et al., *J Am Chem Soc* **2004**, *126*, 3477.
[53] T. K. Ritchie, et al., In Methods in Enzymology; D. Nejat, Ed; Academic Press: New York, **2009**; pp. 211–231.
[54] A. E. Klon, et al., *Biophys J* **2000**, *79*, 1679.
[55] Y. Qi, et al., *J Chem Inform Model* **2014**, *54*, 1003.
[56] Y. Qi, et al., *J Chem Theory Comput* **2015**, *11*, 4486.
[57] X. Cheng, et al., *J Chem Inform Model* **2013**, *53*, 2171.
[58] X. Cheng, et al., *Biochim Biophys Acta (BBA)-Biomembr* **2016**, *1858*, 1566.
[59] S. M. Gruner, et al., *Annu Rev Biophys Biophys Chem* **1985**, *14*, 211.
[60] R. Rand, et al., *Biochemistry* **1990**, *29*, 76.
[61] S. Leikin, et al., *Biophys J* **1996**, *71*, 2623.
[62] Z. Chen, R. Rand, *Biophys J* **1997**, *73*, 267.
[63] N. Fuller, R. Rand, *Biophys J* **2001**, *81*, 243.
[64] S. J. Marrink, A. E. Mark, *Biophys J* **2004**, *87*, 3894.
[65] A. J. Sodt, R. W. Pastor, *Biophys J* **2013**, *104*, 2202.
[66] Y. Qi, et al., *Biophys J* **2015**, *109*, 2012.
[67] Y. Z. Ohkubo, et al., *Biophys J* **2012**, *102*, 2130.
[68] L. Monticelli, et al., *J Chem Theory Comput* **2008**, *4*, 819.
[69] W. Han, K. Schulten, *J Chem Theory Comput* **2012**, *8*, 4413.
[70] S. J. Marrink, A. H. de Vries, A. E. Mark, *J Phys Chem B* **2004**, *108*, 750.
[71] S. J. Marrink, et al., *J Phys Chem B* **2007**, *111*, 7812.
[72] S. O. Yesylevskyy, et al., *PLoS Comput Biol* **2010**, *6*, e1000810.
[73] D. H. de Jong, et al., *J Chem Theory Comput* **2013**, *9*, 687.
[74] C. Arnarez, et al., *J Chem Theory Comput* **2015**, *11*, 260.
[75] X. Periole, et al., *J Chem Theory Comput* **2009**, *5*, 2531.
[76] T. A. Wassenaar, et al., *J Chem Theory Comput* **2014**, *10*, 676.
[77] A. J. Rzepiela, et al., *J Comput Chem* **2010**, *31*, 1333.
[78] P. J. Stansfeld, M. S. Sansom, *J Chem Theory Comput* **2011**, *7*, 1157.
[79] D. W. Borhani, D. E. Shaw, *J Comput-Aid Mol Des* **2012**, *26*, 15.
[80] H. J. Woo, B. Roux, *Proc Natl Acad Sci U S A* **2005**, *102*, 6825.
[81] Y. Deng, B. Roux, *J Chem Theory Comput* **2006**, *2*, 1255.
[82] J. Wang, Y. Deng, B. Roux, *Biophys J* **2006**, *91*, 2798.
[83] Y. Deng, B. Roux, *J Phys Chem B* **2009**, *113*, 2234.
[84] D. L. Mobley, K. A. Dill, *Structure* **2009**, *17*, 489.
[85] H. S. Lee, C. Seok, W. Im, *J Chem Theory Comput* **2015**, *11*, 1255.
[86] S. Jo, et al., *J Chem Inform Model* **2013**, *53*, 267.
[87] D. Beglov, B. Roux, *J Chem Phys* **1994**, *100*, 9050.
[88] W. Im, S. Berneche, B. Roux, *J Chem Phys* **2001**, *114*, 2924.
[89] J. D. Weeks, D. Chandler, H. C. Andersen, *J Chem Phys* **1971**, *54*, 5237.
[90] Y. Deng, B. Roux, *J Phys Chem B* **2004**, *108*, 16567.
[91] W. Jiang, M. Hodoscek, B. Roux, *J Chem Theory Comput* **2009**, *5*, 2583.
[92] W. Jiang, B. Roux, *J Chem Theory Comput* **2010**, *6*, 2559.
[93] S. Jo, et al., *J Comput Chem* **2011**, *32*, 3135.
[94] S. M. Islam, et al., *J Phys Chem B* **2013**, *117*, 4740.
[95] S. M. Islam, B. Roux, *J Phys Chem B* **2015**, *119*, 3901.
[96] B. Roux, S. M. Islam, *J Phys Chem B* **2013**, *117*, 4733.
[97] R. Shen, et al., *PLoS Comput Biol* **2015**, *11*, e1004368.
[98] S. Jo, et al., *Adv Protein Chem Struct Biol* **2014**, *96*, 235.
[99] L. G. Trabuco, et al., *Structure* **2008**, *16*, 673.
[100] R. McGreevy, et al., *Acta Crystallogr Sect D* **2014**, *70*, 2344.
[101] S. Jo, et al., *Nucleic Acid Res* **2008**, *36(WebServerissue)*, W270.
[102] W. Im, D. Beglov, B. Roux, *Comput Phys Commun* **1998**, *111*, 59.
[103] M. Nina, W. Im, B. Roux, *Biophys Chem* **1999**, *78*, 89.
[104] M. Schaefer, M. Karplus, *J Phys Chem* **1996**, *100*, 1578.
[105] T. Lazaridis, M. Karplus, *Proteins* **1999**, *35*, 133.
[106] T. Lazaridis, *Proteins* **2003**, *52*, 176.
[107] M. S. Lee, F. R. Salsbury, Jr, C. L. Brooks, III, *J Chem Phys* **2002**, *116*, 10606.
[108] S. Tanizaki, M. Feig, *J Chem Phys* **2005**, *122*, 124706.
[109] W. Im, M. Feig, C. L. Brooks, III, *Biophys J* **2003**, *85*, 2900.
[110] W. Im, M. S. Lee, C. L. Brooks, III, *J Comput Chem* **2003**, *24*, 1691.
[111] P. Ferrara, J. Apostolakis, A. Caflisch, *Proteins-Struct Funct Genet* **2002**, *46*, 24.
[112] S. A. Hassan, F. Guarnieri, E. L. Mehler, *J Phys Chem B* **2000**, *104*, 6478.
[113] B. Eisenberg, Y. Hyon, C. Liu, *J Chem Phys* **2010**, *133*, 104104.
[114] B. Lu, et al., *J Chem Phys* **2007**, *127*, 135102.
[115] K. I. Lee, et al., *Biophys J* **2011**, *100*, 611.
[116] K. I. Lee, et al., *J Comput Chem* **2012**, *33*, 331.
[117] W. Im, S. Seefeld, B. Roux, *Biophys J* **2000**, *79*, 788.
[118] W. Im, B. Roux, *J Chem Phys* **2001**, *115*, 4850.
[119] P. E. Lopes, et al., *J Chem Theory Comput* **2013**, *9*, 5430.
[120] A. Savelyev, A. D. MacKerell, *J Comput Chem* **2014**, *35*, 1219.
[121] M. J. Field, P. A. Bash, M. Karplus, *J Comput Chem* **1990**, *11*, 700.
[122] B. A. Gregersen, X. Lopez, D. M. York, *J Am Chem Soc* **2003**, *125*, 7178.
[123] Q. Cui, M. Karplus, *J Chem Phys* **2000**, *112*, 1133.
[124] B. C. Knott, et al., *J Am Chem Soc* **2014**, *136*, 8810.

[125] X. Cheng, et al., *Biophys J* **2015**, *108*, 1954.
[126] J. Lee, et al., *J Magn Reson* **2008**, *193*, 68.
[127] D. K. Chakravorty, et al., *J Biomol NMR* **2013**, *56*, 125.
[128] J. Chen, W. Im, C. L. Brooks, *J Am Chem Soc* **2004**, *126*, 16038.
[129] J. L. Knight, C. L. Brooks, *J Chem Theory Comput* **2011**, *7*, 2728.
[130] J. L. Knight, C. L. Brooks, III, *J Chem Theor Comput* **2011**, *7*, 2728.
[131] K. A. Armacost, G. B. Goh, C. L. Brooks, III, *J Chem Theory Comp* **2015**, *11*, 1267.
[132] M. S. Lee, F. R. Salsbury, Jr, C. L. Brooks, III, *Proteins* **2004**, *56*, 738.
[133] J. Khandogin, C. L. Brooks, III, *Biophys J* **2005**, *89*, 141.
[134] J. Khandogin, C. L. Brooks, III, *Proc Natl Acad Sci U S A* **2007**, *104*, 16880.
[135] B. W. Zhang, L. Brunetti, C. L. Brooks, III, *J Am Chem Soc* **2011**, *133*, 19393.
[136] G. B. Goh, J. L. Knight, C. L. Brooks, III, *J Chem Theory Comput* **2012**, *8*, 36.
[137] L. Foit, et al., *Proc Natl Acad Sci U S A* **2013**, *110*, E1254.
[138] G. B. Goh, J. L. Knight, C. L. Brooks, III, *J Chem Theory Comput* **2013**, *9*, 935.
[139] S. M. Law, B. W. Zhang, C. L. Brooks, III, *Protein Sci* **2013**, *22*, 595.
[140] G. B. Goh, et al., *Proteins* **2014**, *82*, 1319.
[141] G. B. Goh, E. N. Laricheva, C. L. Brooks, III, *J Am Chem Soc* **2014**, *136*, 8496.
[142] A. Panahi, C. L. Brooks, III, *J Phys Chem B* **2015**, *119*, 4601.
[143] X. Zeng, S. Mukhopadhyay, C. L. Brooks, III, *Proc Natl Acad Sci U S A* **2015**, *112*, 2034.
[144] M. Yang, A. D. MacKerell, Jr, *J Chem Theory Comput* **2015**, *11*, 788.
[145] M. Yang, J. Huang, A. D. MacKerell, Jr, *J Chem Theory Comput* **2015**, *11*, 2855.
[146] T. Mori, et al., *Biochim Biophys Acta (BBA)-Biomembr* **2016**, *1858*, 1635.
[147] E. L. Wu, et al., *Biophys J* **2015**, *109*, 2090.
[148] E. L. Wu, et al., *Biophys J* **2013**, *105*, 1444.

Received: 13 August 2016
Revised: 4 October 2016
Accepted: 18 October 2016
Published online on 14 November 2016