CHARMMing: A New, Flexible Web Portal for CHARMM

Benjamin T. Miller,[†,‡] Rishi P. Singh,[†,‡] Jeffery B. Klauda,[‡,§] Milan Hodošček,[||] Bernard R. Brooks,[‡] and H. Lee Woodcock III*[,‡]

Laboratory of Computational Biology, National Heart Lung and Blood Institute, National Institutes of Health, Bethesda, Maryland 20892, and Center for Molecular Modeling, National Institute of Chemistry, Hajdrihova 19, SI-1000 Ljubljana, Slovenia

Received April 15, 2008

A new web portal for the CHARMM macromolecular modeling package, CHARMMing (CHARMM interface and graphics, http://www.charmming.org), is presented. This tool provides a user-friendly interface for the preparation, submission, monitoring, and visualization of molecular simulations (i.e., energy minimization, solvation, and dynamics). The infrastructure used to implement the web application is described. Two additional programs have been developed and integrated with CHARMMing: GENRTF, which is employed to define structural features not supported by the standard CHARMM force field, and a job broker, which is used to provide a portable method for using grid and cluster computing with CHARMMing. The use of the program is described with three proteins: 1YJP, 1O1O, and 1UFY. Source code is provided allowing CHARMMing to be downloaded, installed, and used by supercomputing centers and research groups that have a CHARMM license. Although no software can replace a scientist's own judgment and experience, CHARMMing eases the introduction of newcomers to the molecular modeling discipline by providing a graphical method for running simulations.

1. INTRODUCTION

A common lament among novice and more experienced computational biologists is the lack of easy to use tools for preparing structures and setting up simulations. Although a wide variety of graphical and command line driven molecular modeling software exists,[1−6] in most cases successful use requires years of training. The end result is that newcomers to the field, be they students or scientists from other disciplines, face a significant learning curve. While detailed knowledge of the software is ultimately necessary, a more user-friendly interface would help less experienced users become independent. To provide such an interface, a new web-based tool named CHARMMing (CHARMM interface and graphics) is introduced. This new tool provides a friendly interface for the CHARMM (*C*hemistry at *HAR*vard *M*acromolecular *M*echanics)[6] software package and is designed to facilitate the preparation, execution, and visualization of molecular simulations.

CHARMMing is not the first web-based tool developed to facilitate molecular modeling and simulation. For example, perhaps the first web-based tool designed to ease computational work is WebMO, a free interface to numerous computational chemistry packages.[7] The online ProPKA[8] web site (http://propka.ki.ku.dk/ ~ drogers/) allows for quick calculation of protein pK_a values. The ProBis web site (http://tyr.cmm.ki.si/probis/bin/probis.php) allows structural similarity searching in multiple related proteins to find conserved binding regions involved in protein−protein interactions.[9,10] The GLYCAM Biomolecule Builder[11] (http://www.glycam-.com/CCRC/biombuilder/biomb_index.jsp) provides an interactive carbohydrate builder as well as tools to simplify the use of different classes of biomolecules. It facilitates building glycoprotein and glycolipid structure files for use with CHARMM or AMBER. Moreover, it has the ability to generate Sander input files for running molecular dynamics simulations. CHARMM-GUI[12,13] (http://www.charmm-gui-.org) provides a graphical interface for setting up protein and membrane simulations with CHARMM. Tools are supplied that assist with reading in structures, solvating them, setting up molecular dynamics input files, performing PBEQ calculations, and building protein−membrane complexes. Computational Science & Engineering Online (CSEO) also attempts to provide access to a number of molecular modeling tools and facilitate running them in distributed computing environments.[14]

CHARMMing differs from the aforementioned interfaces in three respects: first, unlike ProPKA, it is focused on a wide variety of methods; second, unlike the GLYCAM or CHARMM-GUI tools, it is designed to allow flexible combinations of different methods; and third, unlike CSEO; it is designed to be run entirely through a Java and Javascript enabled web browser with no additional client software required. CHARMMing contains a set of tools for uploading protein structures or retrieving them from the Protein Data Bank (PDB),[15] performing energy minimizations and simulations, solvation, and viewing the results. Calculations can be performed in arbitrary order, thus giving the end user maximum flexibility; for example, results of one operation can be used as input to another. The interface also incorporates a number of freely available tools such as Jmol[16] for

* Corresponding author. E-mail: hlwood@nih.gov.
† Performed equal work.
‡ National Heart Lung and Blood Institute.
§ Current address: Department of Chemical and Biomolecular Engineering, University of Maryland, College Park, Maryland 20742.
|| National Institute of Chemistry.

10.1021/ci800133b CCC: $40.75
Published on Web 08/13/2008

visualization and an automatic residue topology file (RTF) generator (GENRTF),[17] which creates topology and parameter information for residues not currently supported by the CHARMM force field.[18] User supplied topology and parameter files are also supported.

Two additional important and unique features of the current work are the following: (1) This package is open source software, and the community as a whole is encouraged to contribute to its development. (2) Integration with popular queuing systems allows CHARMMing to be installed as a front-end to computational resources in individual research groups and computational centers alike.

Given its flexibility, research groups will find this new tool useful for many tasks:

1. Job preparation. CHARMMing contains a number of tools that parse PDB format files and create CHARMM input scripts. The currently supported functionality includes energy minimization, solvation (implicit[19] and explicit), molecular dynamics, Langevin dynamics, and self-guided Langevin dynamics.[20] Each of these functions allow extensive customization (i.e., adding patches, applying restraints, charge neutralization).

2. Job Submission and Monitoring. Currently the software supports job submission to Condor[21] and PBS[22] (Portable Batch Systems) based (including TORQUE[23] and PBSPRO[24]) queuing systems. Additionally, jobs are actively monitored and the status is updated continuously. As with standard job submission, there is no requirement that users maintain an active session; logging out of the website and returning later will not affect jobs that have been submitted.

3. Visualization of Structures and Trajectories. CHARMMing can illustrate structures that have been uploaded and can create movies from trajectory files generated from dynamics simulations. The primary visualization package used and distributed is Jmol,[16] but similar packages with Javascript application programming interfaces (APIs) can also be integrated. For example, an interface to ChemAxon's MarvinSpace[25] visualization software is also distributed.

4. Testing and Deployment Platform for New Methods. A web-based interface allows new methods to be accessed without forcing users to learn complicated command line syntax. It is also straightforward for those with some software development experience to add functionality from other packages that have an existing interface with CHARMM.

5. Tutorial and Instructional Aid. A tutorial/introduction to the use of CHARMM has been integrated into the site. This tutorial also provides conceptual foundations to molecular simulation techniques such as energy minimization, solvation, molecular dynamics, and the use of periodic boundary conditions that are implemented in CHARMMing. Thus, the web site may be used as an introduction to molecular simulations in general and CHARMM in particular. With a web interface, students can concentrate on learning the techniques as opposed to software commands. For example, the interface guides the student through a simulation and provides pop-up descriptions of the various areas of the site.

Section 2 describes the fundamental design goals and tools used to implement the current application. Section 3 shows examples of various proteins, highlighting CHARMMing's unique features, section 4 describes planned enhancements to the software, and section 5 summarizes the work and gives the URLs for CHARMMing and its source code.

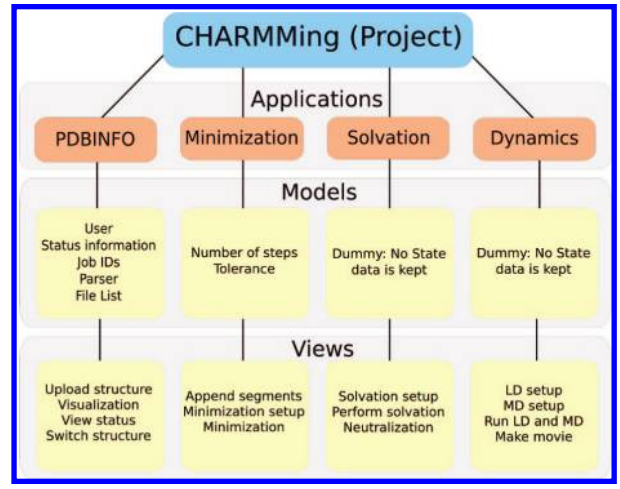

Figure 1. Structure of CHARMMing. The program is divided into several applications, the most important of which are shown here.

2. SOFTWARE DESIGN AND IMPLEMENTATION

2.1. Application Framework. The primary design considerations are flexibility, transparency, and adaptability. The CHARMMing software is built on top of the freely available Django[26] web application framework. The application framework handles various low-level details of web programming so that developers can concentrate on implementing functionality. Django was chosen for its technical capabilities and because it is built around the popular Python programming language. CHARMMing is intended to be used with an up-to-date browser [Mozilla Firefox version 2 or later, Internet Explorer 7 or later, Safari, etc.].

CHARMMing is divided into applications, each of which encapsulate some important part of the program (Figure 1). Each application has its own data model which stores information about the particular task being performed and a set of functions which process commands and are collectively called the application's views. Data generated by one CHARMMing application can be used in another. For example, a protein can be solvated and the resulting structure minimized. The end result of this architecture is a highly flexible and easy to modify code.

To give an example of the internal structure of the program, consider the PDBINFO application. The data model contains the location of data files such as the PDB, CRD (CHARMM card formatted file), and PSF (protein structure file) files, information about how many segments or chains are present, and what operations have been performed. The views contain functions that handle requests to visualize or download files for the structure, parse a newly uploaded PDB into its segments, and switch to a different PDB, among others. New features can be added to provide additional functionality. CHARMMing makes this simple because the support infrastructure is already in place; development need not start from scratch.

One of the benefits of using Django is that it supports multiple independent accounts. Therefore, users can run simulations without interfering with the work of others. It is necessary to register for an account, and Django provides

customizable registration forms; currently, basic contact information and institutional affiliation must be given. Administrators have special accounts that are given access to the Django administration interface. This allows them to modify or delete data stored in the system as well as approve new users.

CHARMMing, sitting on top of Django, must be run on a web server. Although, Django contains its own built-in web server, that is suitable for development and testing, use of the popular Apache[27] web server is recommended. Since Django is designed for developing web applications, it contains a number of features that assist in software debugging. The information given, however, could allow malicious misuse of the web server, and therefore, these features should be disabled on production sites. In this case, errors encountered will appear to the user as nonspecific internal web server errors

2.2. Installation Procedure. A Python script has been developed to facilitate the installation of CHARMMing on a web server running a Linux operating system. As the installer will need to create files and directories as well as install additional software, it is assumed that the installer will be run with administrative privileges. The installer performs the following tasks: (1) install Django (this may be skipped if it is already installed), (2) copy the CHARMMing source distribution and data directories to user-specified locations, (3) set the login name and password for the administrative user, (4) download and install third-party software not distributed with CHARMMing, (5) configure limits to the use of CHARMMing (e.g., the number of dynamics steps allowed), and (6) optionally configure Apache. In cases where the user's intervention is necessary, the install script gives clear directions on what must be done and can be resumed from the stopping point once the required action is taken. This is accomplished by writing a configuration file that is read when restarting installations.

2.3. Automatic Topology and Parameter Generation. *2.3.1. GENRTF.* GENRTF is used to generate the necessary CHARMM input scripts for nonstandard or small molecules not supported by the CHARMM all-atom protein force field.[18] GENRTF reads and analyzes arbitrary Cartesian coordinate files in either PDB, XYZ, MOL, or CHARMM's CRD format. Using the information obtained, input scripts are generated from the standard CHARMM topology file. In the case where an unknown residue is encountered a new topology is created and appended to the RTF. In CHARMMing's implementation, GENRTF is only passed the structural features that are non-native to the CHARMM force field.

For example, using a PDB file, GENRTF creates an input script to generate the native CHARMM coordinate and PSF files. If an unsupported substrate is identified, GENRTF builds the bonding structure for this substrate and creates new topology and parameter information for all such residues. This information can subsequently be read into CHARMM and used for calculations that include the previously undefined parts of the structure. Currently, the code assigns distinct parameters only for undefined atomic elements. While generating the PSF from PDB files, it is convenient that the residue numbering is preserved. Therefore, the original PDB file is separated, one file containing each chain. The generated input script then combines these segments into the complete system. In addition, GENRTF can add the necessary control file and commands and set up ab initio QM/MM calculations using CHARMM's interface to either the Q-Chem or GAMESS quantum chemistry programs.[28−31] In this case, the accuracy of the parameters provided by GENRTF are not a concern as the undefined portion of the structure will likely be treated quantum mechanically. In other cases, generic parameters are provided; a list of these can be found in the Supporting Information (Table 1). Although these parameters are not intended for use in meaningful simulations, they do provide a bridge for two important steps; they serve as a good starting point for parameter refinement and allow CHARMMing to create workable CHARMM compatible files for future use.

In CHARMMing, all of the unknown heteroatoms are separated into their own segment, and a PDB for that segment is created and run through GENRTF. The input file produced is not used directly, instead the CHARMM formatted topology and parameter information is parsed and used to create new RTF and parameter files. Incorporating the newly generated files provides functioning RTF and parameter files in a facile manner that can be subsequently improved.

2.3.2. Antechamber and GAFF. Antechamber is a collection of external programs for automatically determining atom and bond types, evaluating atomic equivalence, generating RTF files, and determining unknown force field parameters.[32] This set of tools has access to the general AMBER force field (GAFF)[33] and can write CHARMM RTF and parameter files which can be subsequently used to create PSF and CRD files. GAFF has parameters for nearly all organic molecules made of C, N, O, H, S, P, F, Cl, Br, and I. The interface between Antechamber and GAFF is implemented to extend the applicability of CHARMMing to molecules or parts of molecules that are not well defined by the standard CHARMM force field.

2.4. Job Broker. In addition to the aforementioned code, CHARMMing includes a standalone program called the "job broker". The purpose of this component is to provide a mechanism to submit jobs to a local or remote computing cluster or grid. There are two parts to the job broker: a front-end which is called upon by the applications and a back-end which interfaces with the grid or cluster queuing system. An illustration of this scheme is presented in Figure 2.

Applications interact with the job broker via a common interface. The advantage of this design is that the job broker need not run on the web server itself, which will be useful at high traffic sites. Currently, submitting new jobs and checking the status of running jobs is supported; however, job termination is a planned feature. If a series of CHARMM scripts needs to be run sequentially, they can be submitted as a single batch and the job broker will guarantee that they are run in order. This is possible because of the directed acyclic graph functionality of CONDOR and job dependencies in PBS, which allow the submission of interdependent tasks. If one of the preliminary jobs fails, the entire operation will be marked as having failed. The job broker may run on the same machine as the web application; however, since all communication with the job broker occurs over the network, this is not a requirement.

The back-end contains all the code that is specific to a particular batch queuing system and is responsible for the actual submission and monitoring of jobs. Back-ends have

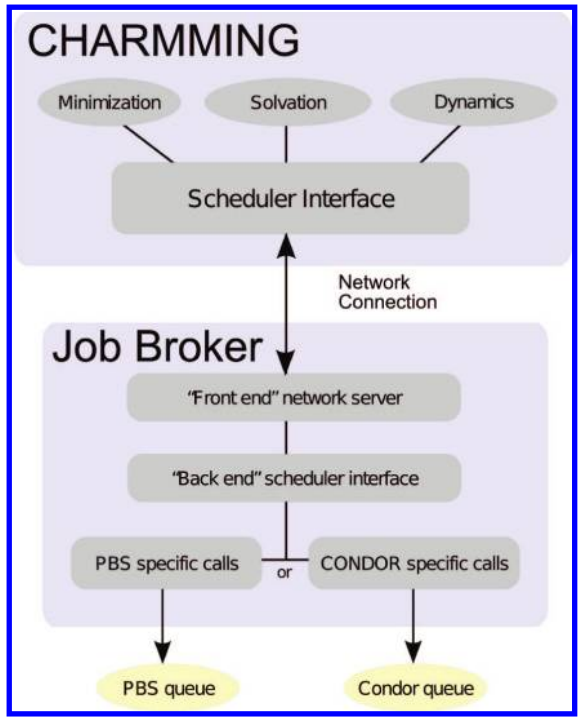

Figure 2. Structure of the job broker. The part of the code that is specific to the queuing system (the back end) is abstracted out, allowing support for different queuing systems to be easily added. Communication between CHARMMing and the job broker takes place over a network connection.

been written for PBS and Condor, which are widely used on traditional supercomputers (i.e., supercomputing centers), workstation clusters, and computational grids. Support for other queuing systems can be added by writing a new back-end. Each back-end can be "plugged in" to the job broker and acts as a wrapper around the normal commands used by the batch scheduler that it supports, e.g. qstat and qsub for PBS queues and condor_q and condor_submit for Condor. Both PBS and Condor support submitting jobs to queues on remote machines; however, when submitting to a remote computer, care must be taken that staging of input and output files is handled correctly, as CHARMMing treats file staging as the responsibility of the batch system. PBS systems can stage files using the standard *NIX rcp or scp commands, and Condor provides a method (using condor_compile) to create executables that will transparently send I/O back to the submission node. It should also be possible to use Condor-G, which uses the Globus cluster toolkit[34] to submit jobs to remote sites; however, this has not yet been tested with CHARMMing. For use with a local cluster, a shared NFS filesystem is probably the easiest method of storing files. CHARMMing allows user data to be stored in an arbitrary location separate from the web site code, which maximizes flexibility when preparing such a configuration.

3. RESULTS AND DISCUSSION

3.1. Preparing a Structure. *3.1.1. Submitting the Structure.* The first task in any project is to obtain a structure in usable format and resolve inconsistencies between PDB format and CHARMM card format. The CHARMMing coordinate parser provides access to a robust input tool. Structures are submitted to the website using the **Submit Structure** form, which is located under the **File Manager** submenu. Users may upload their own PDB, CHARMM formatted protein structure (PSF) and coordinate (CRD) files, amino acid sequence (e.g., "ALA GLY PRO"), or provide a PDB ID and allow the file to be downloaded automatically from http://www.pdb.org.[15] As an advanced feature, custom topology and parameter files can be uploaded to define parts of the structure that are nonstandard.

If the structure obtained is in PDB format, the header of the PDB file will be parsed and basic information such as the name of the structure, the author, and the journal in which it was published will be determined. The information gleaned is displayed with an opportunity to modify it. To illustrate the unique features of the software, we demonstrate calculations on a structure from the yeast prion Sup35, human deoxy hemoglobin, and chorismate mutase from *Thermus thermophilus* (PDB codes 1YJP,[35] 1O1O,[36] and 1UFY,[37] respectively). These proteins are shown in CHARMMing's Jmol visualizer in Figure 3.

3.1.2. Structure Parsing and Appending. Once CHARMMing obtains a structure, a significant amount of work is performed automatically. A diagram of this process is presented in Figure 4. If the file is to be retrieved from the PDB,[15] the crystallographic information file (CIF)[38] is fetched and the CIFtr program[39] converts it to PDB format. The CIF format is used because there is no universal format for PDB files whereas the CIF is well specified. The CIFtr program is thus able to do a reasonable job of producing a PDB file that can be handled by CHARMMing's parser (which uses the formal Protein Data Bank specification). After this step, the PDB is split up into its constituent models (any or all models found in the file may be worked with) and the naming conventions for residues and atoms are fixed by a routine that performs the same functions as the convpdb.pl of MMTSB.[40] The subsequent PDB is split up into its component segments, which consist of three types: protein segments, "good" heteroatoms, and "bad" heteroatoms. The good heteroatoms are defined as those for which the CHARMM protein all-atom force field has topology and parameter data. Examples of good heteroatom types include waters (TIP3P) and heavy metals such as zinc and iron. When unknown (bad) heteroatoms are present, CHARMMing invokes GENRTF to create topology and parameter files. This process is shown in Figure 5. While the parameters given by GENRTF are generic, they provide enough information to conduct basic operations. This is required for 1UFY and 1O1O as both contain substrates not present in the standard CHARMM topology files.

During the initial processing, all residues with multiple protonation states (histidine, lysine, glutamic acid, and aspartic acid) are displayed and the user is given the opportunity to edit them individually. If changes are made to histidine residues, the PDB files are modified to reflect the new sequences. The CHARMM force field does not contain separate topologies for all states that are listed; therefore, a stream file is prepared containing patches that change the protonation states of the selected residues. For convenience, a view of the protonizable residue and its immediate surroundings out to a radius of 8 Å is provided on the **View PDBs** page.

When the first task (e.g., minimization) is requested individual segments must be concatenated. A PSF and CRD file for each desired segment is prepared and the custom

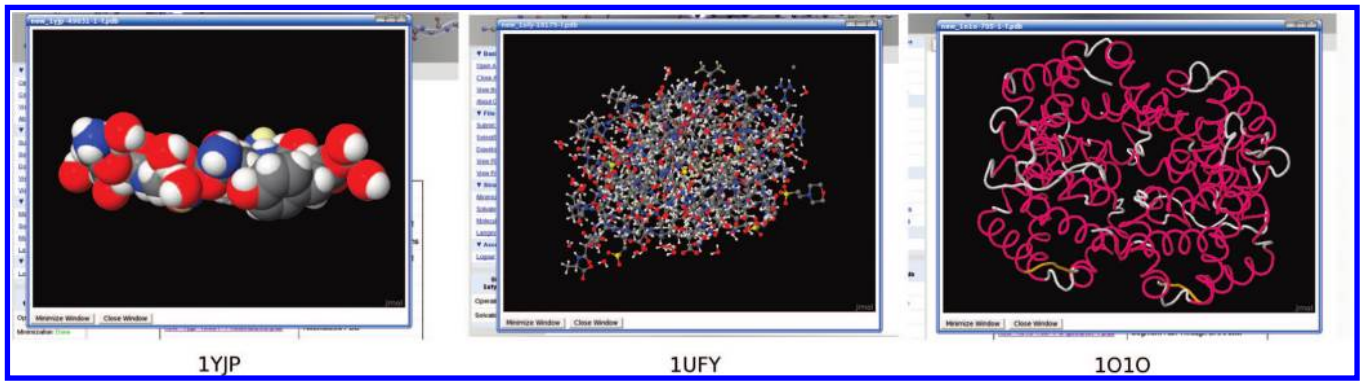

Figure 3. 1YJP (left), a yeast prion, 1UFY (middle), a chorismate mutase, and 1O1O (right), a deoxy Hemoglobin, as shown in CHARMMing's built-in Jmol viewer.

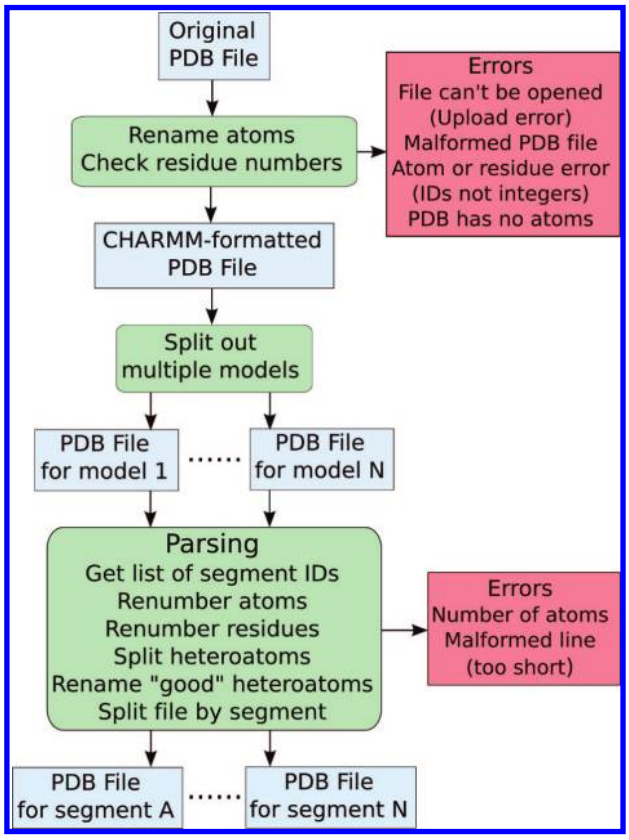

Figure 4. Diagram of the PDB parsing workflow with possible errors highlighted. Files are denoted with square boxes while operations are shown with rounded corners.

protonation stream files are processed on the appropriate segments. Following this process all hydrogen position are rebuilt using CHARMM's *HBUIld* utility. This is an essential step as crystal structures typically do not have hydrogen positions defined and the positions of the hydrogens are dependent upon which segments are combined into the final structure. At this point, final PSF and CRD files for the appended structure are created.

3.1.3. Energy Minimization. Once a structure is prepared, the first step is usually to perform energy minimization to remove any undesired stresses (i.e., bad van der Waals contacts). On the left-hand menu, **Minimize** can be selected from the **Structure/System** submenu to bring up the minimization dialogue (Figure 6). Here, it is possible to select which chain(s) to minimize; in the case of 1YJP, there are two options: the protein itself (segment a) and its CHARMM-compliant heteroatoms (a-goodhet). The latter are crystal

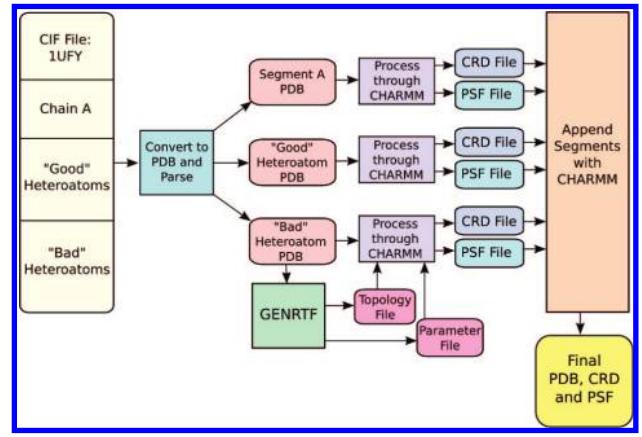

Figure 5. Workflow for structure preparation. When a new PDB file is submitted, it is parsed, unknown residues are passed through GENRTF, PSF and CRD files are produced, and segments are appended together as needed.

waters. Similar segments exist for 1UFY and 1O1O, also representing embedded water.

CHARMMing supports two of the several minimization algorithms available in CHARMM: steepest descent (SD) and the adopted basis Newton−Raphson (ABNR) method, although support for other minimizers can easily be added. From the minimization dialogue, the number of steps for each type of minimization, the gradient tolerance, and additional parameters can be specified. Custom patches (e.g., disulfide bonds) and the SCPISM[19] implicit solvent model may also be applied. To further control the minimization, various portions of the structure can be selected and restrained. An option is also provided to fix heavy atoms and thus only minimize the hydrogen positions. This is often necessary as hydrogens placed by *HBUIld* are often not in optimal positions.

3.1.4. Solvation. Although an implicit solvation model can be used, it is more typical that explicit waters are added to solvate the system before performing molecular dynamics. This can be a difficult process and much effort has been made to simplify this task. As with minimization, the **Solvation** page (Figure 7) is available under the **Structure/System** submenu and allows the selection of segments to be solvated. The shape of the water structure may be chosen from cubic, spherical, hexagonal, and rhombic dodecahedral (RHDO). CHARMMing can also automatically select an appropriate solvation structure for the user. If this option is chosen a check is done on the dimensions of the protein. If the either the shortest or longest axes are more than 30% different from

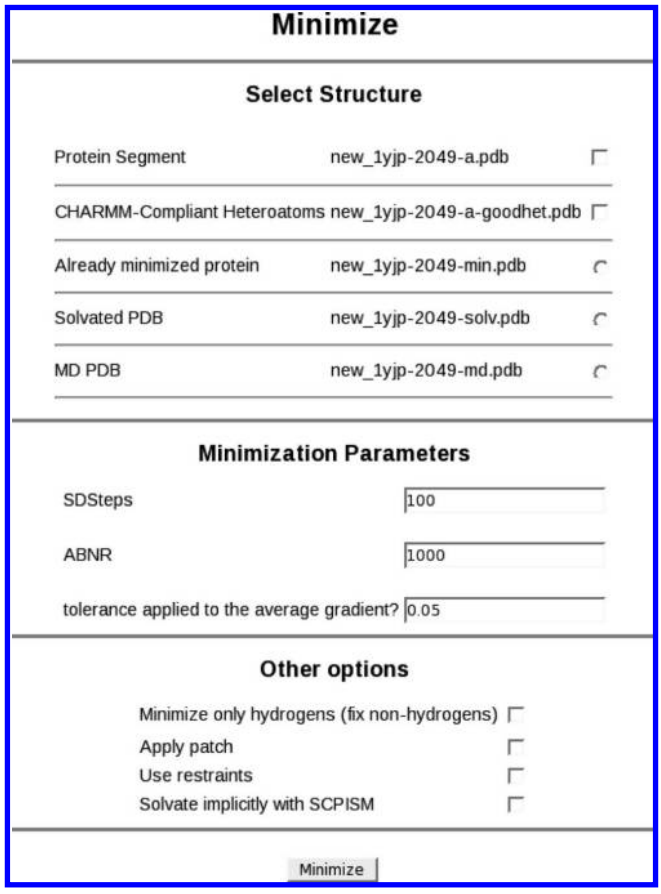

Figure 6. Minimization display.

the median axis, then the system is solvated in a hexagon; otherwise, an RHDO is used. Whenever a hexagon is used, a comment is placed into the CHARMM script reminding the user that it may be necessary to restrain the structure to keep it from rotating outside of the hexagon during long dynamics simulations. CHARMMing currently does not support automatically adding such restraints. The software uses CHARMM's *COORdinate* commands to determine an appropriate dimension for the box based on two possible criteria; how far the water is specified to extend beyond the structure or exact user specified dimensions. Once the form is submitted, an input script is generated that reads in water coordinates and creates the correct geometric shape, deleting waters that overlap with the solute. Note that this procedure can also introduce internal waters within cavities, but not in a measured manner. That is the number of water molecules is likely to be nonoptimal.

An additional option given during the solvation procedure is the ability to automatically neutralize the system by adding counterions. This is a useful procedure if particle-mesh Ewald (PME) is subsequently employed as charge neutrality is desired (in fact, CHARMMing will not allow the user to run a simulation with PME on a non-neutral system because of several problems that may arise).[41] Currently, six ions are supported; potassium, sodium, cesium, magnesium, calcium, and chloride. By default, KCl is used and a concentration of approximately 0.15 M is applied; however, these are user defined parameters, and any combination of positive and negative ions can be used in addition to setting the concentration. For example, 0.15 M KCl is indicative of cellular concentration whereas much lower KCl concentrations are found in the blood stream. Therefore, by making the concentration an adjustable parameter, simulation conditions can easily be varied.

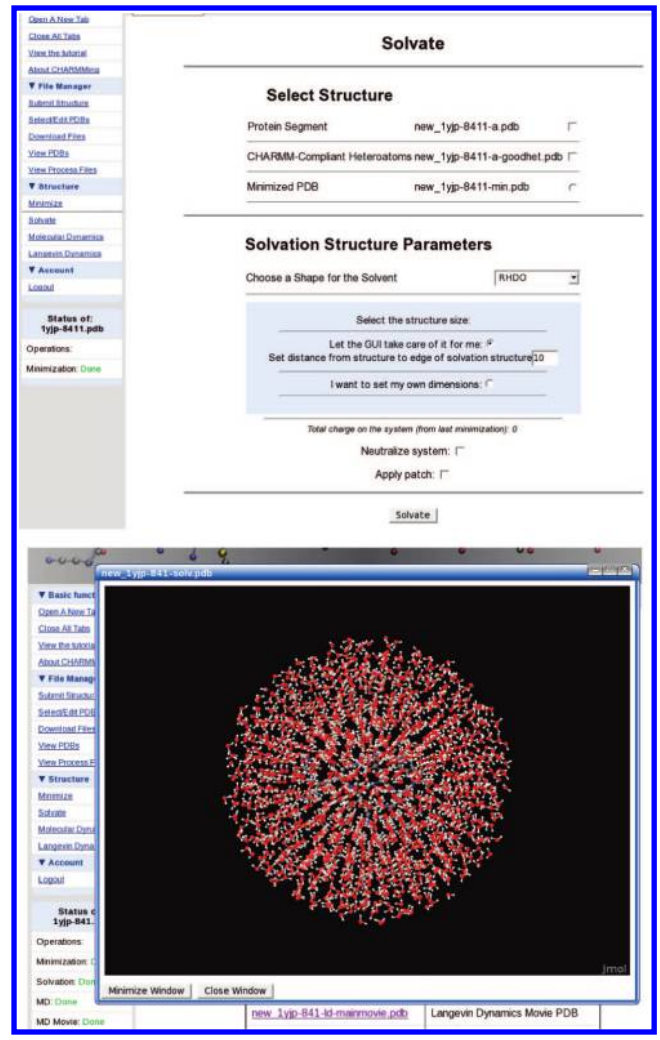

Figure 7. (top) Solvation page, which allows proteins to be solvated in various water structures. (bottom) View of 1YJP solvated in a water ball.

The neutralization procedure is carried out via the following steps: concentration and counterions are defined, the number of ions required to reach that concentration is calculated using the difference between the protein volume and the total volume of the box, and the ions are added via a Monte-Carlo-like procedure where random waters are replaced by ions and the energy (following brief minimization) is compared to the previous step with higher energy conformations being rejected. The number of Monte-Carlo steps is an adjustable parameter in the setup procedure. After solvation and neutralization, it may be necessary to reminimize the structure to remove bad contacts (those with high energy). This is easily accomplished by revisiting the minimization dialogue, selecting the newly solvated structure, and applying periodic boundary conditions to a new minimization.

3.2. Running Dynamics. CHARMMing provides interfaces to both molecular dynamics and Langevin dynamics. Like minimization and solvation, both of the options are available from the **Structure/System** submenu (Figure 8). Periodic boundary conditions for molecular dynamics are employed via CHARMM's *CRYStal* and *IMAGe* commands

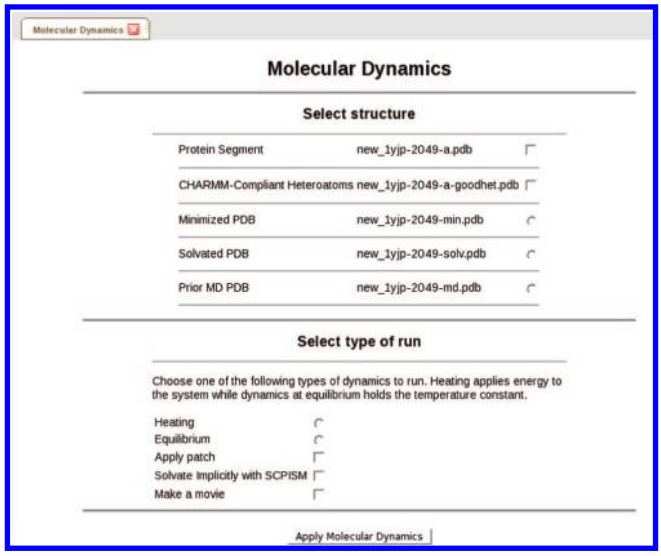

Figure 8. Form used to set up molecular dynamics calculations.

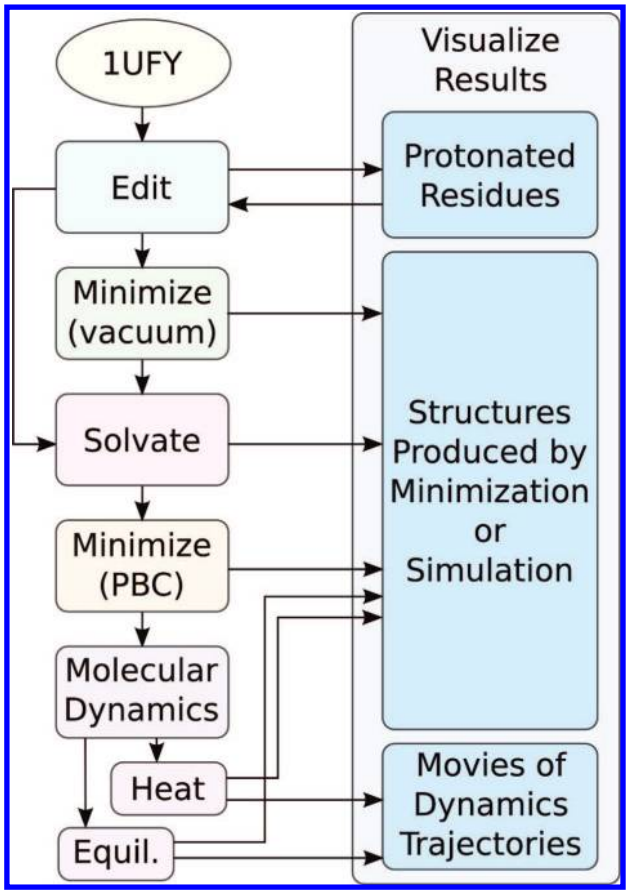

Figure 9. Suggested sequence of steps for running an explicitly solvated molecular dynamics simulations of chorismate mutase from *Thermus thermophilus* (PDB code: 1UFY).

when available (i.e., for all nonvacuum systems except those solvated in a spherical water structure or that employ implicit solvation). This effectively replicates the simulated structure, forming a perfect lattice. This eliminates both vacuum/boundary effects and problems caused by atoms moving outside the unit cell.[42] The suggested progression for running molecular dynamics simulations in explicit solvent is illustrated in Figure 9.

Heating runs, in which energy is gradually added to the system (the frequency and scale of the temperature increases may be configured), or equilibration runs, in which temperature is kept constant at a user specified value, can be specified. Heating is useful for applying thermal energy to a minimized structure while equilibration is a useful precursor for production runs. CHARMM's velocity rescaling functionality is employed to increase system energy in heating runs. Constant temperature and pressure (*CPT*) simulations with the Hoover thermostat[43,44] are used for equilibration runs other than for spherically solvated, vacuum, and implicitly solvated systems. The Langevin piston method[45] is used for all equilibration runs that also employ periodic boundary conditions. The pressure piston mass (*PMASs*) is set to 2% of the molecular weight of the system while the mass of the thermal piston (*TMASs*) is set to 10 times that value. Constant pressure (*PCONs*) is also used for heating nonspherically solvated structures, but *CPT* and the Hoover thermostat are not used as these are inappropriate in this case. For structures that do not use periodic boundary conditions and all heating runs, molecular dynamics with the leapfrog Verlet integrator and no pressure or temperature constraints are used. Currently, molecular dynamics runs are limited to 10 000 steps (with a 1 fs time step, a maximum of 10 ps). This value was chosen to prevent overloading the computational resources dedicated to this project; however, sites that want to use CHARMMing as a full scale front-end can easily increase the limit during the installation procedure.

For systems solvated in a cube, hexagon or RHDO, CHARMM's PME capabilities are employed in molecular dynamics simulations to efficiently calculate electrostatic interactions. An external program is used to determine appropriate fast Fourier transform (FFT) dimensions for the PME summation, and charge neutrality is tested. Finally, the *SHAKe* command is used to fix hydrogen bond lengths for those residues supported by the standard CHARMM force field or a user uploaded force field. Applying *SHAKe* constraints requires that the structure be well-minimized after all hydrogens (including ones in solvent) have been added. Failure to do so may lead to errors and/or failure of the simulation.

For Langevin dynamics, the leapfrog Verlet integrator is used with the *LANGevin* keyword.[46] This adds random forces into the system as specified by the Langevin equation and simulates the presence of solvent drag on the system. The collision frequency is specified via the *FBETa* scalar value. Although periodic boundary conditions are not used, *SHAKe* is still applied as for molecular dynamics. Additionally, the self-guided Langevin dynamics method of Wu and Brooks is available via the *SGLD* keyword.[20] This adds a guiding force to the Langevin equation based on the momentum of the system, causing more conformations to be sampled in a given time period.

Beyond the normal process of running jobs, there is an additional step that must be taken if a movie showing the motion of the system during dynamics is requested. A multiframe PDB is extracted from the dynamics trajectory using CHARMM's trajectory input/output (I/O) mechanism. Jmol is very particular about the format of PDBs used to create movies; therefore, a separate parser exists to combine the coordinates into a properly formatted file. Movies are currently limited to 10 frames, but can easily extended for longer simulations by editing the movie making script.

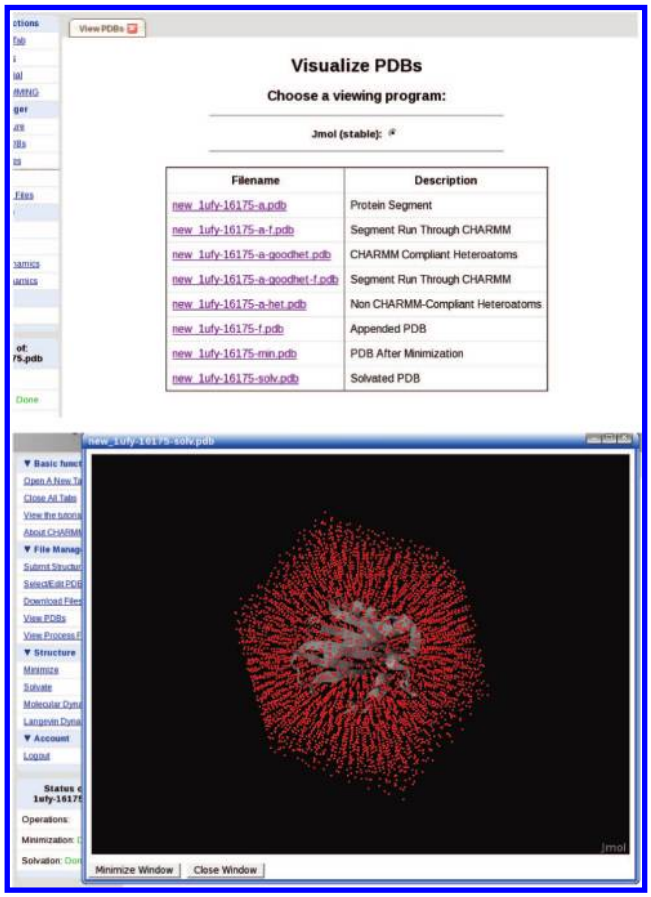

Figure 10. (top) CHARMMing's visualization page. Jmol is available and an interface to ChemAxon has also been developed. (bottom) Result of visualizing 1UFY solvated in a rhombic dodecahedral water structure. The red dots are the water oxygens.

3.3. Viewing and Analyzing Output. CHARMMing tracks submitted jobs in a database and provides a continuously updated status display. Once jobs have completed there are several methods available for accessing the results, all of which are available under the **File Manager** submenu. The input, output, structure, and coordinate files may be downloaded from the **Download Files** page. Files can be downloaded individually or concatenated as a tar archive. The input and output files can also be viewed directly in a web browser via the **View Process Files** page (Figure 10). Even if a job has failed, the input and output files are provided here to facilitate debugging.

Visualization options are available on the **View PDBs** page for runs that have succeeded. Multiple viewing programs, such as Jmol or MarvinSpace, can be selected if installed. The default viewer used in all of the figures is Jmol, which is distributed with CHARMMing. When a PDB is selected from the list, a new window containing the Jmol application appears. By right-clicking within this window, Jmol's standard functionality can be accessed. When a movie PDB is selected, Jmol is set to automatically play it in a loop, although this behavior can be changed.

4. FUTURE PLANS

CHARMMing.org is currently developed and managed by the same research group that manages CHARMM.org, which ensures that the project has active support from notable members of the CHARMM development community. CHARMMing's development community and open source nature give the project staying power and imply that it will not simply disappear or become tied to one research group. In particular, it allows members of the community with diverse research goals to drive development forward. Undoubtedly, future developers will have their own ideas about what functionality is most desirable; however, there are several obvious steps to make it a more useful tool for the community, which are enumerated in this section.

Better small molecule support is a desirable goal for CHARMMing. The software currently supports the CHARMM27 protein force field, which is to be expanded to both CHARMM and third-party force fields (e.g., AMBER, OPLS-AA, etc.). This will allow for easy simulations of systems containing lipids, carbohydrates, and small molecules. However, in cases where small molecules form a vital part of a structure, a hybrid QM/MM treatment is often desired, and it is therefore useful for CHARMMing to support these methods. Fortunately, as mentioned above, CHARMM already contains interfaces to a number of ab initio and semiempirical codes, and it would be relatively straightforward to add an additional Django application to create QM/MM input scripts. A full-fledged implementation of this would take advantage of the GUI and allow users to visually select regions to be modeled at different levels of theory, a feature currently available only in commercial packages. This would require modification to the Jmol viewing engine. Fortunately, Jmol is open source, and therefore can be expanded to meet future requirements.

It is also necessary to add support for different types of simulations. For example, there has recently been a renewal of interest in vibrational analysis methods. A generalized facility for normal-mode analysis would therefore be a useful addition to CHARMMing. CHARMM's *VIBRan* facility already provides the computational engine. This functionality could be combined with all-atom, QM/MM, and coarse grained models such as the Elastic Network Model[47] to allow a multiscale approach to various problems.

CHARMMing can also be improved by adding analysis tools. Although output and trajectory files can be downloaded, it would be desirable for analysis to be done directly through the interface. However, it is impossible to anticipate and implement all of the possible options for analysis. Therefore, a mechanism should be added to allow trusted users to edit the CHARMM scripts directly through the interface or even upload their own scripts. This is not currently permitted due to security concerns (e.g., users could use CHARMM's *SYSTem* command for malicious purposes), but Django's user authorization system could be leveraged to enable an administrator to give only highly trusted users access to script editing features.

Another goal is to use Django's template system to further decouple the application logic from the functionality. Specifically, the CHARMM scripts should be independent of the low-level Python code. This will facilitate the implementation of new features as very little Python programming expertise will be required for developers to make new techniques available. This strategy will also ease integration of new software packages into CHARMMing. For instance, a template based script generation system could be configured

to produce NAMD or GROMACS scripts instead of or in addition to CHARMM scripts.

The end goal of this additional work is to create a complete simulation management system. Currently, a user must click through each of the submission forms in turn. A more efficient system would be to allow them to create a set of jobs that can be submitted together. For example, minimization, solvation, dynamics, and analysis jobs could all be set up at once, and then run in any order, with the user being notified when outputs are available. Such a feature would make CHARMMing more useful as a grid portal.

5. CONCLUSION

CHARMMing, a new graphical portal to CHARMM, has been presented. It is anticipated that this new tool will assist both novice and expert computational biologists in efficiently and effectively setting up and carrying out complex simulations. No software, however, can replace a scientist's in-depth knowledge of both the simulation methods and the tools used to carry them out. For this reason, results obtained from CHARMMing should not be accepted without detailed consideration. Fortunately, by providing easy access to raw input and output files, the software eases this task considerably. Additionally, a 50 page integrated tutorial is included as an introduction to CHARMM which further assists both novice and advanced users.

The software currently supports several different simulation techniques: energy minimization, solvation (implicit and explicit), molecular dynamics, Langevin dynamics, and self-guided Langevin dynamics. For each of these actions, several parameters related to the calculations can be customized. On the basis of this input, CHARMMing will produce and run CHARMM scripts. Back-end computing resources such as Beowulf clusters, grids, and supercomputers can be accessed via PBS or Condor based batch queuing systems. While other web interfaces exist, CHARMMing is well suited for setting up and managing protein simulations. There are detailed plans in place to continue adding features to CHARMMing and expand it to support more complex use cases. CHARMMing's source code is available for download from http://source.charmming.org, and contributions from the community to expand its functionality or fix bugs will be gratefully accepted, as befitting the project's open source nature. The web-based version of CHARMMing can be found at http://www.charmming.org. A default user name and password is listed on the web site, and user accounts can also be created.

ACKNOWLEDGMENT

We wish to acknowledge Richard M. Venable and Xiongwu Wu for their assistance with preparing the CHARMM input scripts. We also wish to thank the developers of Django (an open source web application framework), Jmol (used for visualization), Ajax Tabs (used to provide the tabbed user interface), Bubble tooltips (provides context-sensitive help), Prototype (a Javascript framework), SDMenu (for the side menu), TableKit (For table sorting), and GreyBox (for various display functionality). This research was supported by the Intramural Research Program of the NIH, NHLBI.

Supporting Information Available: Technical details describing the architecture of CHARMMing highlighting the web applications, scheduler daemon, and client side environment. Additionally, the bond, angle, and torsional parameters assigned by GENRTF are listed. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Humphrey, W.; Dalke, A.; Schulten, K. VMD−Visual Molecular. *Dynamics J. Mol. Graphics* **1996**, *14*, 33–38.
(2) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.
(3) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435–447.
(4) Sayle, R. A.; Milnerwhite, E. J. RASMOL - Biomolecular graphics for all. *Trends Biochem. Sci.* **1995**, *20*, 374–376.
(5) Case, D. A.; Cheatham, T. E., III; Darden, T.; Gohlke, H.; Lou, R.; Merz, K. M., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.
(6) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. CHARMM - A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4*, 187–217.
(7) WebMO. http://www.webmo.net (accessed May 30, 2008). WebMO is a free world wide web-based interface to computational chemistry packages.
(8) Li, H.; Robertson, A. D.; Jensen, J. H. Very fast empirical prediction and rationalization of protein pK(a) values. *Proteins* **2005**, *61*, 704–721.
(9) Konc, J.; Janezic, D. Protein-protein binding-sites prediction by protein surface structure conservation. *J. Chem. Inf. Model.* **2007**, *47*, 940–944.
(10) Carl, N.; Konc, J.; Janezic, D. Protein surface conservation in binding sites. *J. Chem. Inf. Model* **2008** *48*, 1279.
(11) Glycam Biomolecule Builder. http://www.glycam.com/CCRC/biombuilder/biomb_index.jsp (accessed May 30, 2008). The Glycam Biomolecule Builder is a free, online builder for carbohydrates and related molecules (e.g., glycoproteins).
(12) Jo, S.; Kim, T.; Im, W. Automated builder and database of protein/membrane complexes for molecular dynamics simulations. *PLoS One* **2007**, *2*, e880.
(13) Jo, S.; Taehoon, K.; Iyer, V. G.; Im, W. CHARMM-GUI: A web-based graphical user interface for CHARMM. *J. Comput. Chem.* **2008**, in press.
(14) Truong, T. N.; Nayak, M.; Huynh, H. H.; Cook, T.; Mahajan, P.; Tran, L. T.; Bharath, J.; Jain, S.; Pham, H. B.; Boonyasiriwat, C.; Nguyen, N.; Andersen, E.; Kim, Y.; Choe, S.; Choi, J.; Cheatham, T. E.; Facelli, J. C. Computational science and engineering online (CSE-Online): A cyber-infrastructure for scientific computing. *J. Chem. Inf. Model.* **2006**, *46*, 971–984.
(15) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The protein data bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.
(16) Steinbeck, C.; Han, Y. Q.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The chemistry development kit (cdk): An open-source java library for chemo- and bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493–500.
(17) Hodoscek, M. GENRTF.http://a.cmm.ki.si/genrtf (accessed May 30, 2008). GENRTF is a program to convert cartesian coordinates into CHARMM input files.
(18) MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. A* **1998**, *102*, 3586–3616.
(19) Hassan, S. A.; Guarnieri, F.; Mehler, E. L. A general treatment of solvent effects based on screened Coulomb potentials. *J. Phys. Chem. B.* **2000**, *104*, 6478–6489.
(20) Wu, X. W.; Brooks, B. R. Self-guided langevin dynamics simulation method. *Chem. Phys. Lett.* **2003**, *381*, 512–518.
(21) Thain, D.; Tannenbaum, T.; Livny, M. Distributed computing in practice: the Condor experience. *Concurrency: Pract. Exper.* **2005**, *17*, 323–356.

(22) OpenPBS. http://www.openpbs.org (accessed May 30, 2008). OpenPBS is the original version of the Portable Batch System, a queuing system developed for NASA in the early 1990s.
(23) TORQUE. Cluster Resources, Inc. http://www.clusterresources.com/pages/products/torque-resource-manager.php (accessed May 30, 2008). TORQUE is an open source resource manager providing control over batch jobs and distributed compute nodes.
(24) PBS Professional. Altair Inc. http://www.pbsgridworks.com (accessed May 30, 2008). PBS Professional is a service orientated architecture and field-proven grid infrastructure software.
(25) ChemAxon. http://www.chemaxon.com (accessed May 30, 2008). Marvin 4.1.11, 2007, was used for drawing, displaying, and characterizing chemical structures, substructures, and reactions.
(26) Lawrence Journal-World. Django. http://www.djangoproject.com (accessed May 30, 2008). Django is a framework that makes it easier to build better world wide web applications more quickly and with less code.
(27) Apache Software Foundation. Apache HTTP Server Project. http://httpd.apache.org (accessed May 30, 2008). The Apache HTTP server an open-source HTTP server for modern operating systems including UNIX and Windows NT.
(28) Shao, Y.; Molnar, L. F.; Jung, Y.; Kussmann, J.; Ochsenfeld, C.; Brown, S. T.; Gilbert, A. T. B.; Slipchenko, L. V.; Levchenko, S. V.; O'Neill, D. P.; Distasio, R. A.; Lochan, R. C.; Wang, T.; Beran, G. J. O.; Besley, N. A.; Herbert, J. M.; Lin, C. Y.; Van Voorhis, T.; Chien, S. H.; Sodt, A.; Steele, R. P.; Rassolov, V. A.; Maslen, P. E.; Korambath, P. P.; Adamson, R. D.; Austin, B.; Baker, J.; Byrd, E. F. C.; Dachsel, H.; Doerksen, R. J.; Dreuw, A.; Dunietz, B. D.; Dutoi, A. D.; Furlani, T. R.; Gwaltney, S. R.; Heyden, A.; Hirata, S.; Hsu, C.-P.; Kedziora, G.; Khalliulin, R. Z.; Klunzinger, P.; Lee, A. M.; Lee, M. S.; Liang, W.; Lotan, I.; Nair, N.; Peters, B.; Proynov, E. I.; Pieniazek, P. A.; Rhee, Y. M.; Ritchie, J.; Rosta, E.; Sherrill, C. D.; Simmonett, A. C.; Subotnik, J. E.; Woodcock, H. L.; Zhang, W.; Bell, A. T.; Chakraborty, A. K.; Chipman, D. M.; Keil, F. J.; Warshel, A.; Hehre, W. J.; Schaefer, H. F.; Kong, J.; Krylov, A. I.; Gill, P. M. W.; Head-Gordon, M. Advances in methods and algorithms in a modern quantum chemistry program package. *Phys. Chem. Chem. Phys.* **2006**, *8*, 3172–3191.
(29) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S. J.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. General atomic and molecular electronic-structure system. *J. Comput. Chem.* **1993**, *14*, 1347–1363.
(30) Woodcock, H. L.; Hodoscek, M.; Gilbert, A. T. B.; Gill, P. M. W.; Schaefer, H. F.; Brooks, B. R. Interfacing Q-Chem and CHARMM to perform QM/MM reaction path calculations. *J. Comput. Chem.* **2007**, *28*, 1485–1502.
(31) Eurenius, K. P.; Chatfield, D. C.; Brooks, B. R.; Hodoscek, M. Enzyme mechanisms with hybrid quantum and molecular mechanical potentials.1. theoretical considerations. *Int. J. Quantum Chem.* **1996**, *60*, 1189–1200.
(32) Wang, J. M.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic atom type and bond type perception in molecular mechanical calculations. *J. Mol. Graphics Modell.* **2006**, *25*, 247–260.
(33) Wang, J. M.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and testing of a general amber force field. *J. Comput. Chem.* **2004**, *25*, 1157–1174.
(34) Foster, I. Globus toolkit version 4: Software for service-oriented systems. *J Comput. Sci. Technol.* **2006**, *21*, 513–520.
(35) Nelson, R.; Sawaya, M. R.; Balbirnie, M.; Madsen, A. O.; Riekel, C.; Grothe, R.; Eisenberg, D. Structure of the cross-beta spine of amyloid-like fibrils. *Nature* **2005**, *435*, 773–778.
(36) Brucker, E. A. Genetically crosslinked hemoglobin: a structural study. *Acta Crystallogr., Sect. D.* **2000**, *56*, 812–816.
(37) Crystal analysis of chorismate mutase from thermus thermophilus 1UFY. http://www.pdb.org(accessed May 30, 2008).
(38) Hall, S. R.; Allen, F. H.; Brown, I. D. The crystallographic information file cif - a new standard archive file for crystallography. *Acta. Crystallogr., Sect. A* **1991**, *47*, 655–685.
(39) Feng, Z.; Westbrook, J. CIFTr. http://sw-tools.pdb.org/apps/CIFTr/index.html (accessed May 30, 2008). CIFTr is an application program for translating files in mmCIF format into files in PDB format.
(40) Feig, M.; Karanicolas, J.; Brooks, C. L. MMTSB tool set: enhanced sampling and multiscale modeling methods for applications in structural biology. *J. Mol. Graphics Modell.* **2004**, *22*, 377–395.
(41) Bogusz, S.; Cheatham, T. E.; Brooks, B. R. Removal of pressure and free energy artifacts in charged periodic systems via net charge corrections to the ewald potential. *J. Chem. Phys.* **1998**, *108*, 7070–7084.
(42) Cheatham, T. E.; Brooks, B. R. Recent advances in molecular dynamics simulation towards the realistic representation of biomolecules in solution. *Theor. Chem. Acc.* **1998**, *99*, 279–288.
(43) Nose, S. A unified formulation of the constant temperature molecular-dynamics methods. *J. Chem. Phys.* **1984**, *81*, 511–519.
(44) Hoover, W. G. Canonical dynamics - equilibrium phase-space distributions. *Phys. Rev. A* **1985**, *31*, 1695–1697.
(45) Feller, S. E.; Zhang, Y. H.; Pastor, R. W.; Brooks, B. R. Constant-pressure molecular-dynamics simulation - the langevin piston method. *J. Chem. Phys.* **1995**, *103*, 4613–4621.
(46) Allen, M. P.; Tildesley, D. J. *Computer Simulations of Liquids*; Oxford Science Publications: Oxford, NY, 1987.
(47) Tirion, M. M. Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys. Rev. Lett.* **1996**, *77*, 1905–1908.

CI800133B