JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

A Flexible, Grid-Enabled Web Portal for GROMACS Molecular Dynamics Simulations

Marc van Dijk,[†] Tsjerk A. Wassenaar,[‡] and Alexandre M.J.J. Bonvin[†,*]

[†]Bijvoet Center for Biomolecular Research, Faculty of Science—Chemistry, Utrecht University, Padualaan 8, 3584 CH, Utrecht, The Netherlands

[‡]Groningen Biomolecular Sciences and Biotechnology Institute, Rijksuniversiteit Groningen, Nijenborgh 7, 9747AG, The Netherlands

ABSTRACT: Molecular dynamics simulations are becoming a standard part of workflows in structural biology. They are used for tasks as diverse as assessing molecular flexibility, probing conformational changes, assessing the impact of mutations, or gaining information about molecular interactions. However, performing a successful simulation requires sufficient computational resources, familiarity with the simulation software, and experience in the setup of a system and the analysis of the resulting trajectories. These considerations become especially critical in large-scale parametric MD simulations. Offering such tools to a wide user community requires a robust and versatile, but user-friendly, facility for molecular dynamics simulations with access to vast computational resources. Here, we present the GROMACS grid-enabled Web portal for the setup and execution of molecular dynamics simulation on the WeNMR grid infrastructure, a distributed network of computational resources within the European Grid Initiative. The Web portal aims at ease-of-use through automated setup of the simulation system using best-practice protocols, yet allowing for tuning of key parameters. Alternatively, the simulation can be started from preconfigured GROMACS simulation systems. Performing multiple lengthy calculations using multiple processors on the WeNMR grid infrastructure ensures scalability. The combination of analysis routines for quality assurance and automatic recovery in case of failures provides a reliable platform for MD simulations. The GROMACS Web portal is embedded within the services of the WeNMR Virtual Research Community (VRC) accessible from http://www.wenmr.eu/wenmr/nmr-services. It is freely accessible upon registration with a valid X509 personal certificate with the enmr.eu Virtual Organization (VO).

INTRODUCTION

Molecular dynamics (MD) simulations of macromolecules have come a long way since the pioneering work performed some four decades ago.[1] They have grown to become a standard tool for complementing experiments, providing a structural basis for rationalizing in vitro and in vivo observations, and for suggesting new experiments. Advances in algorithms[2] and hardware[3,4] have allowed ever larger systems to be simulated for ever longer times, providing, among other things, exciting views on function-related dynamics and assembly of full virus particles,[5] protein folding events and mechanisms,[6−8] and long time scale dynamics.[9−11]

Very large systems and long time scales often stir the imagination most but often remain an academic exercise limited to a small number of dedicated expert research groups with access to supercomputer resources. On the other hand, large-scale parametric studies form an exciting development. These involve typically many simulations that are combined afterward and allow for studies such as equilibrium state analysis and biomolecular interaction and affinity analysis. Such parametric studies typically do not require massive parallel supercomputer resources, as is the case for the state-of-the-art examples mentioned above. Still, the computing requirements may well surpass the local resources available to (experimental) groups aiming at complementing their work with MD simulations. Large-scale studies also ask for robust and standardized procedures for setup, simulation, and data analysis.

The need for a robust and easy to use solution to the above requirements has led to the development of the GROMACS grid-enabled Web portal presented here. The Web portal automates many of the steps required for setup and execution of MD simulations on the WeNMR grid infrastructure[12] using the popular GROMACS (Groningen Machine for Chemical Simulation)[13,14] MD software. The grid infrastructure, a distributed network of computational resources within the European Grid Initiative (EGI, http://www.egi.eu), provides extensive computational resources particularly suitable for large-scale parametric studies where many simulations can be performed simultaneously. The Web portal aims at ease-of-use and scalability through the automated setup of the simulation system using best-practice protocols and robust management of lengthy simulations. Key parameters in the automatic setup of the simulation system can be configured. Alternatively, preconfigured simulation systems using a GROMACS binary run input file (.tpr file) and an optional checkpoint file (.cpt file) can be used. All of these features are available from a Web browser anywhere in the world. The combination of a user-friendly Web-based interface, automated system setup with support for manual configuration, automatic recovery from simulation failure, and access to ample computational resources via grid access should make the GROMACS Web portal a

Special Issue: Wilfred F. van Gunsteren Festschrift

Received: February 5, 2012

   dx.doi.org/10.1021/ct300102d | *J. Chem. Theory Comput.* XXXX, XXX, XXX−XXX

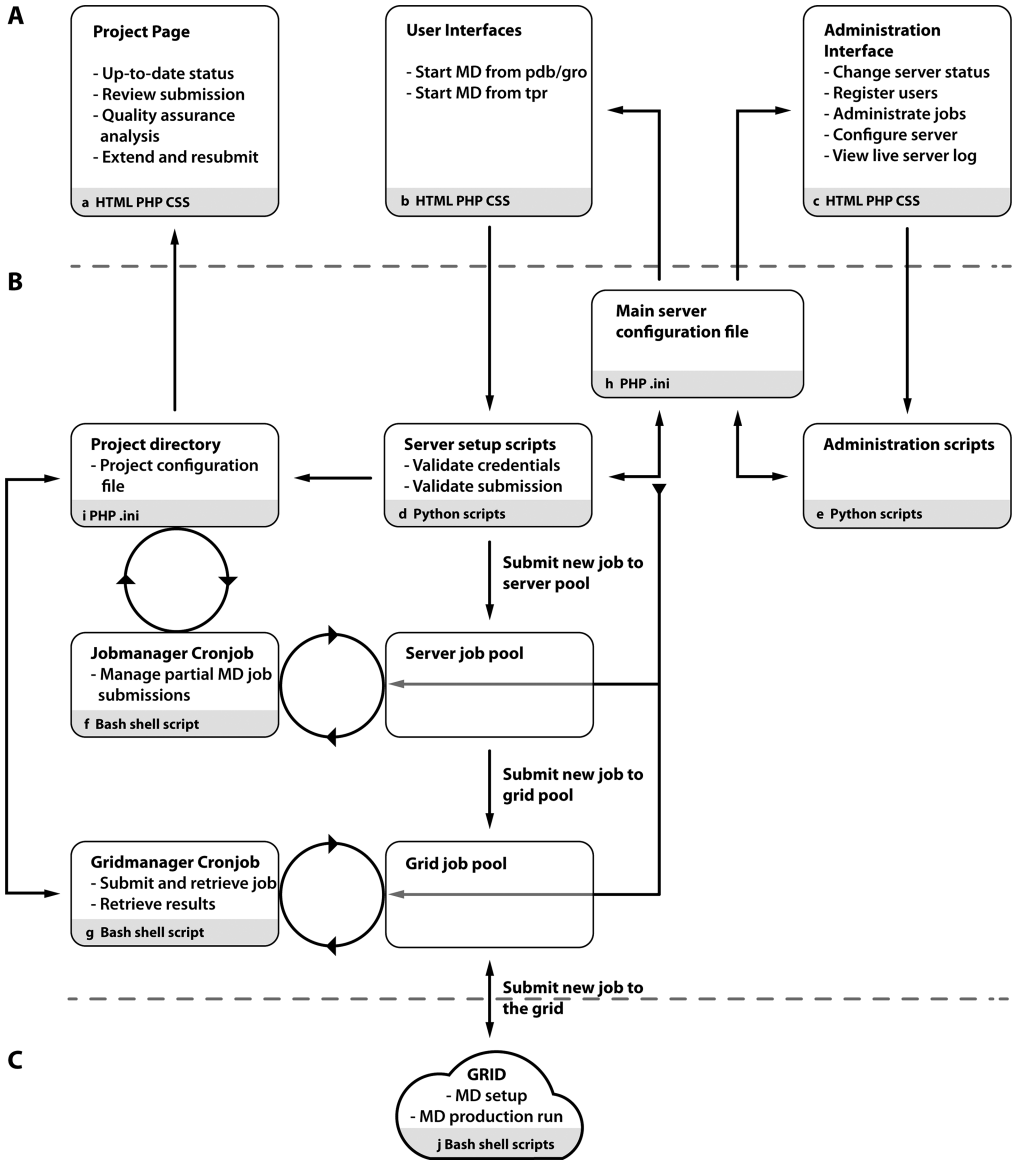

Figure 1. Schematic overview of the GROMACS Web portal architecture divided into three layers: (A) the user Web-based interfaces, (B) the server-based request processing and workflow management scripts, and (C) the grid-based molecular dynamics setup and execution scripts. The technologies used to drive the various parts are shown in the gray boxes.

valuable tool both for parametric MD simulations as well as for researchers who might be unfamiliar with MD simulations and are considering their use in their own work. The portal is freely accessible to nonprofit users worldwide as an embedded service in the WeNMR Virtual Research Community portal at http://www.wenmr.eu.[15] The portal does however require registration with the enmr.eu Virtual Organization (VO) based on a personal X509 certificate. Instructions on how to obtain these credentials are available on the WeNMR Web site.

METHODS

The architecture of the GROMACS Web server consists of three separate layers (Figure 1). The first are the Web pages that allow the user to interact with the server (Figure 1A). The second layer is a set of management scripts that handles the user's requests, validates the input files, prepares a project, submits simulation jobs to the European Grid Infrastructure, and manages the simulation workflow until it is completed (Figure 1B). The third layer (Figure 1C) consists of two MD scripts that are executed on remote grid sites where the GROMACS software has been preinstalled. These scripts handle the preparation and equilibration of the simulation system, as well as the final production simulation. The first two layers represent the server itself, which runs on a dedicated Linux server equipped with the Apache HTTP server software and accessible via a Web browser. These three layers will be discussed in more detail below.

GROMACS Web Portal Front End. The GROMACS Web portal front end (Figure 1A) is a combination of HTML (http://www.w3.org), CSS (http://www.w3.org), and PHP (http://www.php.net) driven Web pages that allow the user to interact with the server. The main Web page provides access to two server interfaces allowing one to

- perform a MD simulation starting from a supplied Protein Data Bank (PDB;[16] .pdb) or GROMACS (.gro) coordinate file (Figure 2A)
- start a simulation from a GROMACS run binary input file (.tpr) and optional checkpoint file (.cpt) (Figure 2B)

dx.doi.org/10.1021/ct300102d | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

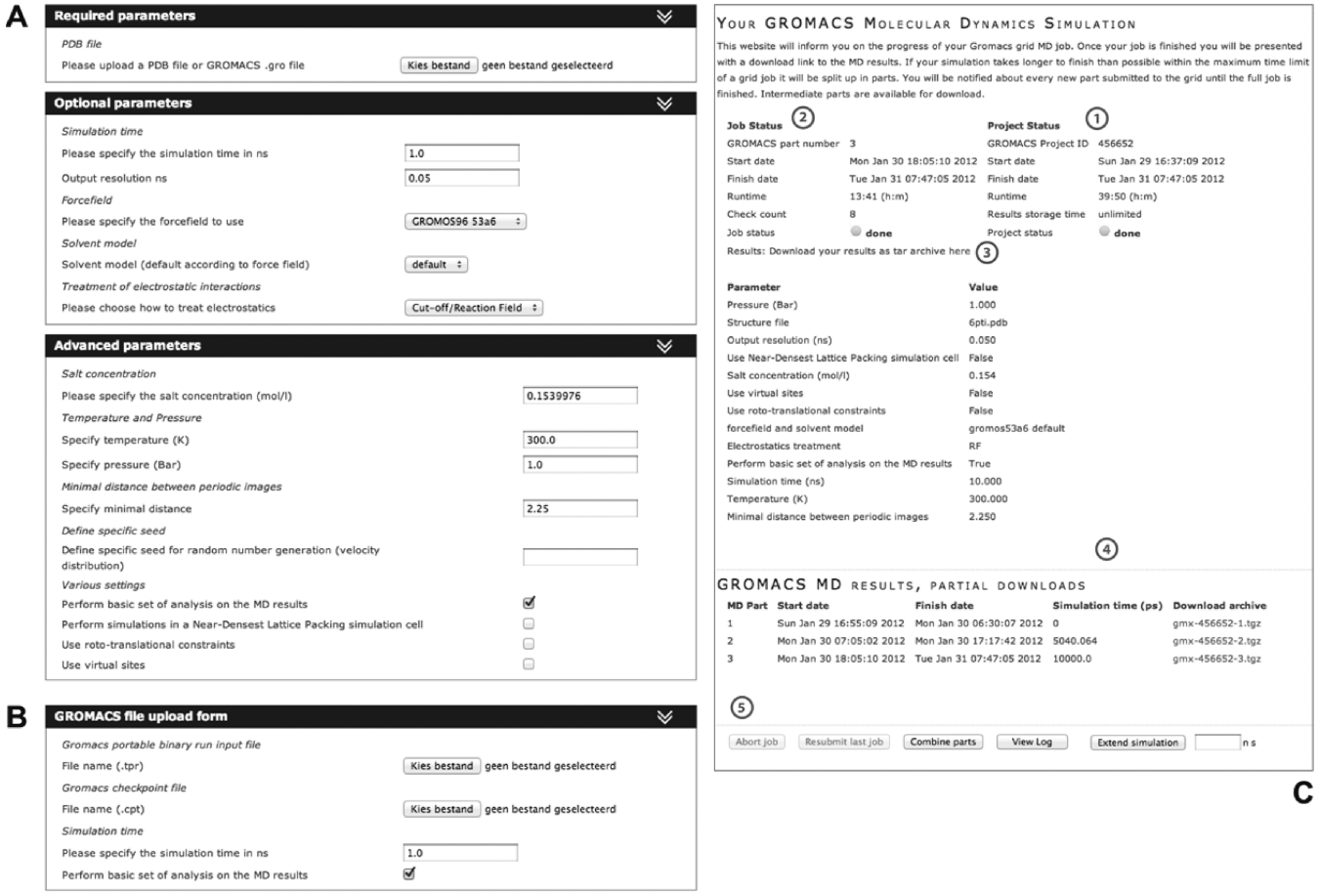

Figure 2. GROMACS user interface Web pages showing (A) the Web form for starting a MD simulation from a PDB (.pdb) or GROMACS (.gro) structure file, (B) the Web form for starting MD simulation from a GROMACS binary run input file (.tpr) and optional checkpoint file (.cpt), and (C) an example MD project page of a finished simulation. The project page shows the status of the project as a whole (1), the simulation part running at that moment (2), a download option for tar-zipped archives of the combined final trajectory (3), every completed simulation part (4), and a task bar to interact with the project (5).

The two interfaces use a foldable menu layout similar to many other WeNMR Web portals to clearly separate minimal required parameters from optional and advanced ones. Data validation upon project submission to either of the two interfaces usually takes less than a minute, after which the user is redirected to a project page (Figure 2C). The information flow between the Web pages (Figure 1, a–c), the request processor (Figure 1, d), and the workflow handlers (Figure 1, f,g) is controlled via table-like databases written to be compatible with the PHP configuration file (.ini) markup. There is one main database that stores the server's configuration (Figure 1, h) and a separate database file for every individual MD project (Figure 1, i) which stores the project's configuration and progress. This setup enables up-to-date information on the progress of a simulation to be displayed on the project page. Furthermore, the project page allows the user to

- resubmit the last simulation snapshot in case of a failure
- combine and optionally analyze all simulation parts generated up to that moment
- extend a completed simulation by a defined number of nanoseconds
- inspect the project log file

The server has a separate password-protected interface aimed at server administrators, which provides an overview of the server's statistics (number of queued, active, and total served projects), basic server settings, and a live view of the server's logs. It also allows one to manage user registration, change the permissions settings, and view statistics on all projects with the ability to inspect, abort, or delete them if needed.

Project Management. Submission of a new MD project through either of the two server interfaces will trigger the request processor Python script (Figure 1, d) that will collect and process the server input. After user authentication, the collected data are validated using the Python-based Spyder framework (www.spyderware.nl) also used in the HADDOCK[17] and CS-ROSETTA[15] Web servers (described in more detail in the Results and Discussion section). Inconsistencies will be reported directly through the browser. After validation, a project directory with a unique ID will be created and populated with the input files, a job database file (.ini), and a grid submission file written using the Job Definition Language (JDL). The JDL file is required to route the job to the proper grid site that meets the requirements to run the job. An example of a JDL file produced by the server is shown in Figure 3.

The execution of the newly created project is managed by two Bash shell scripts: a job manager and grid manager script. Both scripts run as a periodic cron process (Figure 1, f and g, respectively). The job manager copies the required project job

dx.doi.org/10.1021/ct300102d | J. Chem. Theory Comput. XXXX, XXX, XXX–XXX

```
JobType = "Normal";
Requirements = (other.GlueCEPolicyMaxWallClockTime >= 450  && other.GlueCEPolicyMaxCPUTime >= 2700  &&
                Member("VO-enmr.eu-GROMACS4.5.3-rtc-
r2",other.GlueHostApplicationSoftwareRunTimeEnvironment));
Rank = (other.GlueCEStateFreeJobSlots > 100 ? other.GlueCEStateFreeJobSlots : other.GlueCEStateWaitingJobs == 0 ?
        other.GlueCEStateFreeJobSlots * 100 / (other.GlueCEInfoTotalCPUs + 1) :  (-other.GlueCEStateWaitingJobs * 4 / (
        other.GlueCEStateRunningJobs + 1 )) - 1 );
FuzzyRank = true;
CPUNumber = 6;
SMPGranularity = 6;
RetryCount = 0;
ShallowRetryCount = 3;
Executable = "gmx45mdrun.sh";
Arguments = "-np 6 -tpr gmx- ###.tpr -cpt gmx-###.cpt -time 10.0 -maxh 18000 -archive gmx-###.tgz";
StdOutput = "gmx-###.stdout";
StdError = "gmx-###.stderr";
InputSandbox = {"gmx45mdrun.sh", "gmx-###.tpr", "gmx-###.cpt"};
OutputSandbox = {"gmx-###.tgz","gmx-###.stdout","gmx-###.stderr"};
```

Figure 3. Example of a GROMACS Web portal generated .jdl file (Job Definition Language) required for submission and routing of a MD job to a suitable grid site. Proper grid sites are identified on the basis of the following JDL arguments. *Requirements*: specifying respectively the wall clock time in minutes and CPU time in seconds required to run the job, and the requirement for GROMACS version 4.5.3-rtc to be installed at a given site. *Rank*: in case of multiple grid sites matching the requirements, they are ranked in order based on the number of available idle CPUs, the number of jobs waiting in the queue, and the estimated time a job should start running at the site. The *FuzzyRank* argument takes a stochastic approach to selecting a designation site from the ranked sites and contributes to jobs being distributed more evenly over the available sites. The *CPUNumber* argument is required to support multiple CPU cores being available to the threaded execution of the GROMACS mdrun process, and the *SMPGranularity* argument requests a node providing that minimum number of CPU cores. ### indicates a unique, randomly generated number identifying the project.

files to a designated grid pool directory, while the grid manager takes care of submitting new jobs to the grid, updating the status of already submitted jobs, and retrieving the results of completed ones. The grid manager is designed to communicate with the grid using gLite, the middleware distribution used by the EGI grid infrastructure. The modular architecture of the server should allow communication with other grid middleware (e.g., Globus), provided a dedicated grid manager is used instead and the job syntax is adapted.

The grid manager script is equipped to deal with typical grid-related problems, such as interrupted connections or unexpected downtime, by performing the submission and results retrieval process a preset number of times before tagging a job as "failed". Furthermore, if needed, a job submission is attempted at different Workload Management Systems (WMS) on the grid to deal with inaccessible WMS machines. The submission script updates the status of a job in its database file as it progresses from "pending" to "submission", "running", "done", "aborted", or "failed" in case something went wrong. Finally, the grid manager processes cancellation requests from the server.

Changes in the status of a job are processed by the job manager, whose main tasks are to process retrieved results and prepare new grid jobs. The script notifies the user by e-mail for every major change in the status of the project and subsequently updates the projects database to reflect those changes on the projects results page. The post processing script examines the content of the results to assess whether the simulation has run for the specified time or was ended prematurely. In the latter case, the script attempts resubmitting a follow-up simulation part with the input (.tpr) and checkpoint (.cpt) files from the previous simulation part, extending it automatically as long as needed to reach the requested simulation time. The length of the longest queues available at grid sites and the lifetime of the grid proxy certificate limit the time that each simulation job is allowed to run on a grid site. In practice, most jobs are terminated after 24 h. Therefore, the GROMACS mdrun executable is not allowed to run more than a predefined number of hours (wall clock time), depending on the number of requested processors for multithreading. The mdrun −maxh option is used to define this maximum duration. When the full simulation has finished, the job manager script combines all MD trajectories and performs some predefined quality assurance analysis, if requested by the user. This analysis involves

- the evolution of various energy terms as a function of time
- root-mean-square (RMS) deviations from the starting structure as a function of time
- backbone RMS fluctuations from the average structure
- the minimal distance between periodic images as a function of time
- the radius of gyration as a function of time
- secondary structure elements as a function of time

The results of this analysis are presented graphically on the project Web page.

Grid-Based MD Protocol Script. At the back end, running on the grid worknodes, are two Bash shell scripts (Figure 1, j). The first contains a complete and automated MD workflow (Figure 4), whereas the second only covers the production simulation (Figure 4, step 8). Using a separate script for the production run facilitates running simulations in parts. The main script was initially developed as a protocol for a large scale setup of simulations,[18] up to the point where a production run input file was created, which could subsequently be transferred to a cluster. Since then, it has grown to become a flexible and complete wrapper for setting up and performing simulations as a single process, which has made it very suitable for porting to a grid or Cloud environment. The full simulation protocol comprises eight steps, shown in Figure 4. The first seven steps involve setting up and preparing the system (the first script):

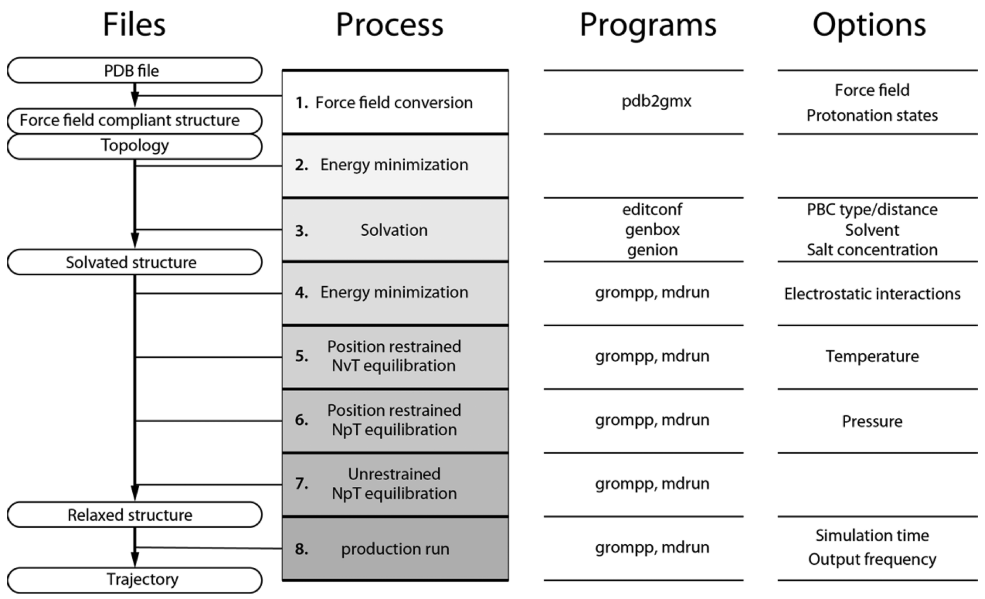

Figure 4. Schematic overview of the seven step protocol used by the MD setup script to prepare the simulation system starting from a PDB or GROMACS structure file. For each of the seven steps denoted as "process", the used GROMACS executable (programs), the available options, and the generated files are shown. The eighth step is taken over by the second script for production simulations that enables the long production simulation to run in parts.

1. checking the input structure, converting it to match the chosen force field, and generating the topological description
2. energy minimization in vacuum
3. setting up the simulation box, solvating the system with an explicit solvent model, and adding Na^+ and Cl^- ions to balance charges and match a given salinity, by default set to physiological concentration (0.154 mol/L)
4. energy minimization with position restraints on the solute with a force constant of 1000 kJ mol^{-1}, nm^{-2}
5. 20 ps equilibration of the thermostat (NVT) and of the solvent, using position restraints on the solute with a force constant of 1000 kJ mol^{-1}, nm^{-2}
6. 3 × 20 ps equilibration of the manostat (NPT), using position restraints on the solute with decreasing force constants of 1000, 100, and 10 kJ mol^{-1}, nm^{-2}
7. 20 ps of unrestrained MD using the same setup as the production simulation; the production simulation follows this step

The setup of the simulation system is followed by a short 20 ps MD with production simulation conditions, which is handled by the second MD script. MD protocols usually diverge mainly with respect to the chosen force fields and simulation parameters. Concerning the latter, it is worth noting that most force fields are developed using a specific set of parameters; these parameters can thus often be regarded as an integral part of the force field. Consequently, the choice of a force field usually defines the parameters for the protocol to be used (Table 1); this forms the basis for the choices made internally in the master protocol. However, the default choice for the treatment of electrostatics and the solvent model can be overruled. Whereas force fields may be parametrized with a shifted cutoff (e.g., OPLS[19]), or a reaction-field correction[20] (e.g., GROMOS[21,22]), some advocate that an Ewald summation method,[23,24] e.g., PME, be used at all times for the treatment of electrostatics to avoid artifacts at the cutoff. This is explicitly available as an option for running the protocol. Other

Table 1. Force Field Families and Their Variants Supported by the GROMACS Web Portal, Each with Their Respective Default Solvent Model and Treatment of Electrostatics

force field family	variants	default solvent model	default treatment of electrostatics
GROMOS 96[21]	43a1, 43a2, 45a3, 53a5, 53a6	SPC[27]	twin-range cutoff (0.9/1.4 nm) with reaction field correction
AMBER[28]	03, 94, 96, 99, 99 SB, 99 SB-ILDN, GS	TIP3P[29]	particle-mesh Ewald, order of 4, 0.125 fourier spacing
CHARMM 27[30]		TIP3P[29]	shifted Coulomb interaction (force switching) shifted between 1.0 and 1.2 nm
OPLS-AA[31]		TIP4P[29]	particle-mesh Ewald, order of 4, 0.125 fourier spacing

parameters that can be controlled directly are the salt concentration, the temperature, and the pressure. By default, the periodic boundary conditions are set to a rhombic dodecahedron with a distance between periodic images of 2.25 nm. However, it is also possible to run simulations in a "molecular shaped" (NDLP) box,[25] yielding simulation systems that are on average more than 50% smaller, and thus runs at least twice as efficiently as the corresponding rhombic dodecahedron. The distance between the periodic images is another parameter that can be set explicitly.

Internally, the MD setup script can be divided into three sections. The first part involves handling of arguments and defining the environment in terms of variables and functions. The second part involves setting up a solvated system, starting from a solute structure. The last part consists of a series of simulations for equilibration and production. Of the three, the second part is the most intricate, as it includes steps that usually require manual intervention. The script offers a command line interface that resembles that of GROMACS, allowing only named options. Due to changes in the syntax between versions of GROMACS, the script is linked to a specific version, which is

dx.doi.org/10.1021/ct300102d | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

4.5.3 for the GROMACS grid Portal (the version of GROMACS deployed at the time of writing on the grid compute elements (CEs)). The use of NDLP periodic boundary conditions furthermore requires that GROMACS be patched to have an implementation of the roto-translational constraints.[26]

The MD production run script is responsible for performing the actual simulation, based on the run input (.tpr) and checkpoint (.cpt) files generated by the MD setup script or uploaded to the server. The server is configured to run production simulations on six CPU cores on designated grid working nodes using a multithreading enabled GROMACS mdrun executable. The GROMACS routines used in the MD setup script use only one CPU core.

The MD production run script is designed to deal with simulations that are run in parts. Internally, the script will assess the total simulation time required based on the .tpr file and the simulation time already performed based on the .cpt file. This information is collected at the start of the script to extend the total simulation time if needed and after the mdrun process exits (regardless of the exit code). This information is stored in a file read by the server to assess the progress of the simulation.

The MD script responsible for performing a given simulation is submitted to the grid by the server together with all other required input files. This setup separates the execution of the MD simulation from the Web server that acts primarily as a project manager. This setup could be easily modified to allow other MD applications to run on the grid, provided suitable workflow scripts are available and the required software has been deployed on the grid infrastructure.

Examples. The use of the GROMACS Web server was demonstrated using PDB[16] structure files from the GROMACS test-set version 4.0.4 (http://www.gromacs.org/Downloads/Test-Set) consisting of the first model of the NMR solution structure of the Alzheimer's disease amyloid A4 peptide[32] (PDB ID: 1aml), the immunoglobulin binding domain of protein G[33] (PDB ID: 1pga), and the bovine pancreatic trypsin inhibitor[34] (PDB ID: 6pti). Each structure was simulated for 10.0 ns using the default GROMOS96 53a6 force field and SPC solvent model. Electrostatic interactions were calculated using a twin-range cutoff with reaction field correction. The simulation systems were prepared for the production run by the MD setup script as described in detail above.

RESULTS AND DISCUSSION

We have developed a user-friendly, grid-based Web portal for running MD simulations of biomolecular systems using the GROMACS software. The portal allows for easy setup of simulations for simple protein systems but also allows running complex systems (e.g., proteins with cofactors or nucleic acids) provided a GROMACS run binary input file (.tpr) is uploaded. The portal is freely accessible to all nonprofit users worldwide upon registration with a valid X509 personal certificate with the enmr.eu Virtual Organization (VO). The WeNMR Web site (http://www.wenmr.eu) provides guidelines on how to obtain these credentials. A GROMACS Web portal account is created after validation of these requirements by the portal administrators. Free access to the grid infrastructure is currently made possible by the support of the EGI and of the national grid initiatives of Belgium, France, Italy, Spain, Germany, The Netherlands (via the Dutch BiG Grid project), Portugal, Spain, United Kingdom, South Africa, Taiwan, and the Latin America GRID infrastructure via the Gisela project. This list will hopefully grow in the future.

The use and performance of the GROMACS Web portal is illustrated using three protein structures from the GROMACS software test set (see Methods section). The process of submitting and managing a new simulation project will be discussed in more detail below.

Submitting a New GROMACS MD Project. The main page of the GROMACS Web portal provides access to two Web server interfaces: the first (Figure 2A) allows starting an MD simulation from a PDB (.pdb) or GROMACS (.gro) structure file, and the second (Figure 2B) allows starting the MD simulation from a pregenerated GROMACS simulation system defined by a binary run input file (.tpr) and an optional checkpoint file (.cpt) in case the simulation needs to be continued from a previous time checkpoint.

The first interface was used for the three example cases. The creation of a new GROMACS MD project using this interface results in the MD setup and production run scripts (Figure 1C) being executed on the grid to prepare and run the simulation system following the eight defined, automated steps described in the Methods section and illustrated in Figure 4. Although this setup procedure is automated, the interface does allow for the following key parameters to be configured:

- *Simulation time*: the simulation time and the output resolution of the trajectory files defined in nanoseconds. To enable fair use of the available grid resources, a maximum simulation time limit is enforced in combination with a size limit on the system and the number of simultaneous active projects. These user policies are described below
- *Force field and solvent model*: Fourteen different force field variants are supported. Each force field has a default solvent model associated with it, as shown in Table 1, although it is possible to explicitly set a different solvent model to be used. The GROMOS96 53a6[21] force field with SPC solvent model[27] is offered as a default by the Web portal
- *Treatment of electrostatic interactions*: By default, the treatment of electrostatic interactions is set to match the settings under which the force field was derived (see Table 1). However, it is possible to choose that electrostatic interactions be computed using the Particle-Mesh-Ewald method[24,35]
- *Salt concentration*: Sodium or chloride ions will be added to compensate any net charge on the solute. In addition, sodium and chloride ions can be added to match a defined concentration, which is set to physiological salt concentration (0.1536 mol/L) by default
- *Temperature and Pressure*: The temperature and pressure are weakly coupled to external baths. For the temperature, the improved Berendsen[36] coupling (v-rescale)[37] method is used, whereas the pressure is coupled using the Berendsen manostat.[36] The temperature is set to 300 K by default but can be controlled through the submission page. The same holds true for the pressure, which is set to 1.0 bar by default. The coupling time used is 0.1 ps for the temperature and 0.3 ps for the pressure
- *Simulation cell and system constraints*: By default, simulations are performed in a rhombic dodecahedron unit cell with a distance between periodic images of 2.25 nm. The distance can be changed on the submission

page. Alternatively, the Near-Densest Lattice Packing option allows for the construction of a minimal-volume simulation box, which can speed up the simulation by about a factor of 2, depending on the system,[25] without noticeable influence on the dynamics of the solute.[18] When using a tight-fitting simulation cell, a roto-translational constraint algorithm is used that prevents reorientations that could lead to violation of the minimal distance between periodic images. For simulations in a rhombic dodecahedron, the use of the roto-translational constraints is optional

- *Virtual sites:* The use of virtual sites[38] offers another possibility to increase the efficiency of the simulation. These are used to replace hydrogen atoms, removing the fastest vibrations in the system that otherwise limit the time step that can be used. Simulations run with virtual sites can therefore use a time step of 5 fs, instead of the 2 fs that is used for simulations with explicit hydrogen atoms

The above parameters are provided as command line arguments to the MD setup script that runs on the grid. This script is the first in the three stages of a full simulation described in detail in the Methods section:

1. preparation of the simulation system using the MD setup script
2. performing the production simulation usually divided into several parts depending on the requested simulation time
3. combination of the trajectories of the simulation and optional quality assurance analysis

While the first Web portal interface performs all three steps, the second interface only performs the last two, starting a simulation from a .tpr and .cpt file using the same MD production script. If a checkpoint file returned by the grid indicates that the production run has not yet finished, the simulation will be extended from the last known time checkpoint. If the simulation job sent to the grid has run to completion, the simulation time as specified in the interface will be used to extend the current simulation. User restrictions with respect to simulation time and system size also apply to this interface. After upload of a structure file and optional definition of the simulation parameters, the data are transferred to the Web server for processing.

Server-Side Processing and User and Project Management. After Web form submission, the uploaded data are first validated. This involves checking user credentials and system parameters. Besides the user name and password, the user credentials involve a restriction on the maximum allowed simulation time in nanoseconds, the number of days the simulation results are stored on the server, and the number of simultaneous simulations a user may perform at any given time. To ensure a fair share of grid resources and storage space on the server, restrictions are imposed on the basis of three user privilege groups, "easy" (10 ns, 5 jobs), "expert" (20 ns, 10 jobs), and "guru" (50 ns, 50 jobs) each with an increasing maximum simulation time and number of active jobs, respectively. New users will be granted "easy" access privileges by default and may be promoted to higher privileges upon request. Validation of system parameters involves a sensible range of defined numerical parameters, consistency of the uploaded structure files, detection of nonstandard residues for which no topology and parameter files are available, and an estimation of the system size by residue count. Structure files that exceed 1000 residues are not allowed. If the validation stage fails, the user is notified immediately via the Web page with the cause of the failure and advice on how to proceed, where possible. The various server and user-based settings can easily be changed using the secure online administration pages.

GROMACS version 4.5.3 deployed on the grid sites supporting the enmr.eu VO is equipped with molecular topology definitions for the most commonly used amino acids, nucleotides, and small organic molecules. However, the ability of the Web portal to start a MD simulation from a GROMACS binary run input file (.tpr) does allow users to upload a prebuilt simulation system containing nonstandard molecules. In this case, the user is responsible for preparing the simulation system. Web portals such as PRODRG,[39] Automated Topology Builder (ATB),[40] SwissParam,[41] and AnteChamber[42] can be used to generate topology and parameter files for nonstandard molecules.

Validation is usually performed in less than a minute and, if successful, results in a new project directory being generated with a unique ID. This directory contains all files required for grid submission. The user is redirected to the Web page belonging to the new project, which displays up-to-date information about the progress of the simulation and a summary of the data used (Figure 2C). A single MD project is usually composed of multiple individual jobs corresponding to various stages of the simulations. The project page provides an up-to-date overview of the status of the project as a whole (Figure 2C,1) and of the simulation part running at that moment (Figure 2C,2). The project status is displayed as *"running"* as long as simulations are running on the grid. It automatically changes to *"processing"* when combining trajectories and performing optional quality assurance analysis. Once all steps have completed, the status changes to *"done"*.

With long MD simulations in particular, it can be helpful to perform an intermediate analysis to assess the quality of the simulation by some quality metrics such as energy and RMSD values as a function of time. To enable this analysis, the project page provides a task bar (Figure 2C,4) that allows the user to interact in real time with the project and perform a few common tasks. The *"Combine parts"* button will combine the trajectories of all simulation parts generated up to that moment and perform a quality assessment analysis if requested. The results of the analysis will be displayed as a number of graphs on the project page. The combined results can be downloaded as a tar-zipped archive. If it appears that the simulation is unsuccessful, it can be aborted using the *"Abort job"* button to free job slots for the user and free grid resources.

The *"Check count"* value in the project status (Figure 2C,2) indicates the number of times the server still has to recover from failures to communicate with the grid or resubmit failed jobs. If the check counter hits 0, the projects status will change to *"aborted"*. This does not mean that the simulation was unsuccessful but could mean that the server had to recover too often from communication- or grid-related problems. In the latter case, the user is encouraged to download the tar-zipped archive of the last successful simulation part (Figure 2C,3) to check if the simulation has run correctly. If so, then the *"Resubmit last job"* button can be used to resubmit the last successful job and reset the check counter to continue the simulation. When the complete simulation has finished, the results are stored on the server for a predefined number of days (7 by default, Figure 2C,2 *"Results storage time"*). If it appears

dx.doi.org/10.1021/ct300102d | J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Table 2. Simulation Results and Server Statistics for the Three Test Cases Used to Validate the Performance of the GROMACS Web Portal

				run times (hours)			efficiency (%)		
PDB ID	size[a]	parts[b]	Mdrun[c]	TPT[d]	TGT[e]	TST[f]	TMT[g]	TGT/TPT[h]	TST/TGT[i]
6pti	15460	6	$10.3_{4.5}$	40.75	37.33	21.37	21.33	92	57
1aml	20973	7	$7.6_{3.4}$	45.92	37.77	29.45	29.41	82	78
1pga	12394	5	$12.2_{6.5}$	22.50	21.53	15.83	15.82	96	73

[a]Total number of atoms in the final simulation system (protein + solvent + ions). [b]The number of subjobs sent to the grid to complete the total 10 ns MD trajectory. [c]Average efficiency of the GROMACS mdrun process expressed in nanoseconds per day calculated over all parts. Values are obtained from the mdrun log files. Standard deviation in subscript. Note that the processor performance of various grid sites does vary. [d]Total Project Time: the total time in hours from project start until the finished simulation, excluding analysis. [e]Total Grid Time: the total time in hours the simulation parts spend on the grid from submission until retrieval. [f]Total Script Time: the total time in hours the simulation scripts were running on the grid nodes for every simulation part. [g]Total Mdrun Time: the total time in hours the GROMACS mdrun process spent simulating the parts. Values are obtained from the mdrun log files. [h]Web portal efficiency: the percentage of the total project time spent processing on the grid. [i]Grid efficiency: the percentage of the total grid time spent processing the jobs on the grid nodes.

that the system was not simulated long enough, the user can extend the simulation using the *"Extend simulation"* option in the task bar.

While the project is running, the status of every simulation part is displayed (Figure 2C,2) together with the start date, the runtime, and the part number. The key changes in the lifecycle of a simulation part are

- *Pending*: the grid job is waiting on the server for submission to the grid
- *Submitted*: the grid job was submitted to the grid and is scheduled to run at a grid site matching the requirements
- *Running*: the grid job is running at a grid site (also communicated by e-mail)
- *Analyze*: grid job results were retrieved from the grid, and their consistency is being checked
- *Done*: the full simulation was successfully finished (also communicated by e-mail)
- *Aborted*: the grid for some reason aborted the current grid job; the server might make a recovery attempt
- *Failed*: the server was unable to perform the simulation due to a simulation error returned by GROMACS or because the maximum number of recovery attempts due to aborted jobs was reached (also communicated by e-mail)

Example Cases. The simulation results for the three protein test cases used to illustrate the performance of the server are shown in Table 2. The test cases were simulated for 10 ns with default conditions (see the Methods section) using six CPU cores via multithreading for the production runs. The table lists the values for the total project time (TPT, from project submission until completion of the simulation excluding analysis), the total time the simulation parts spent on the grid (TGT, from submission until retrieval), the total time spent running the grid-based scripts (TST), and the total time spent by the GROMACS mdrun process performing the actual simulation (TMT). The difference between TPT and TGT is a measure of efficiency for the Web portal and grid-based processes. The Web portal shows good efficiency for the three test cases, indicating that the simulations were processed automatically. Efficiency is likely to drop if user interaction for resubmission of simulation parts is required, or if the grid resources are heavily used.

The difference between TST and TGT is a measure of the grid latency that is influenced by the availability of grid resources. The last simulation part for test case 6pti was resubmitted nine times as a result of grid-related problems before it started running, resulting in a significantly lower efficiency compared to the other cases. Once a grid job starts running at a site, the overhead of the MD scripts on the effective simulation performed by the GROMACS mdrun process (difference between TST and TMT) is marginal and accounts for a few minutes at most.

The grid latency is most responsible for the overall efficiency of the Web portal. The difference in efficiency tends to be correlated with the number of simulation parts required to perform the full simulation. This makes sense, because there is an overhead associated with grid submission, retrieval, and analysis of each simulation part. Simulation parts that were resubmitted because they were aborted or failed will further contribute to a decrease in efficiency. Apart from the system size and simulation time, the number of simulation parts depends on the maximum wall clock time the GROMACS mdrun process is allowed to run, which is set to 5 h for the example cases. Increasing this value will decrease the number of partitions and the time overhead associated with it. However, the grid resource requirements associated with a longer runtime will reduce the number of available sites and therefore not necessarily result in an improvement of efficiency. The simulation efficiency, expressed in the number of nanoseconds simulated per day, depends on the CPU efficiency and number of parallel MD threads at a given grid site. Efficiency might be improved by increasing the number of threads the mdrun process is allowed to allocate, but care must be taken that the number of available grid sites that match these requirements does not decrease as this might reduce the overal efficiency. The default of six threads was found to be optimal for the current infrastructure, since a large number of sites do offer eight core nodes while the number of sites with more than eight cores drops dramatically. Requiring only six out of eight cores increases the chance of finding an empty slot in a working node, which can be a job scheduling concern on grid sites. Next to using threads, GROMACS can be compiled to use MPI (Message Passing Interface) as an option to scale to a larger number of processors. However, the grid is not the ideal medium for such a type of large parallel computations since we do not have control over the communication between nodes, and successful execution might depend on local installation settings and libraries. We chose to avoid such potential problems by using a statically compiled version of GROMACS enabled with multithreading.

CONCLUSIONS

The GROMACS grid-enabled Web portal presented here offers the ability to perform advanced protocolized molecular dynamics simulation using the WeNMR grid infrastructure, a distributed network of computational resources within the European Grid Initiative supported by national Grid Initiatives. The Web portal aims at ease-of-use through automated setup of the simulation system using best-practice protocols, yet allowing for tuning of key parameters or starting the simulation from preconfigured GROMACS simulation systems. Performing multiple lengthy calculations using multiple processors ensures scalability. The combination of analysis routines for quality assurance and automatic recovery from failed simulations when possible provides a reliable platform for MD simulations, which is made accessible to a wide user community with the hope of broadening the use of such simulations to new or less experienced user communities. At this time, the automatic setup is only supporting proteins and DNA without cofactors. However, the upload option allows running complex systems by uploading a prebuilt and equilibrated system, making the server also attractive to more experienced users.

The availability of the GROMACS Web portal through the WeNMR Virtual Research Community (VRC, http://www.wenmr.eu), the largest VRC in the life sciences, should ensure both a tight interaction with the community and the sustainability of the portal. The WeNMR VRC provides a wide range of NMR and structural biology related software tools in addition to the GROMACS Web portal presented here. It not only facilitates access to these tools but also provides tutorials, wikis, and a user help center to guide and assist users in the use of the portals. Finally, it is worth mentioning the recently published MDWeb portal[43] that allows setting up MD systems but does not use the grid infrastructure for production. The availability of such portals should lower the barrier to the use of molecular dynamics simulations by a broad research community.

AUTHOR INFORMATION

Corresponding Author

*Phone: +31.30.2533859. Fax: +31.30.2537623. E-mail: a.m.j.j.bonvin@uu.nl.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by the WeNMR project (European FP7 e-Infrastructure grant, contract no. 261572, www.wenmr.eu). The national Grid Initiatives of Belgium, France, Italy, Germany, The Netherlands (via the Dutch BiG Grid project), Portugal, Spain, U.K., South Africa, Taiwan, and the Latin America GRID infrastructure via the Gisela project are acknowledged for the use of computing and storage facilities. The European Grid Initiative (www.egi.eu) is acknowledged for its support of the WeNMR Virtual Research Community.

DEDICATION

We dedicate this article to Wilfred van Gunsteren on his 65th birthday. It has been a real pleasure to interact and collaborate over the years.

REFERENCES

(1) McCammon, J. A.; Gelin, B. R.; Karplus, M. *Nature* **1977**, *267*, 585−590.
(2) Schlick, T.; Collepardo-Guevara, R.; Halvorsen, L. A.; Jung, S.; Xiao, X. *Q. Rev. Biophys.* **2011**, *44*, 191−228.
(3) Shaw, D. E. Millisecond-scale molecular dynamics simulations on Aton. In *SC ' 09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, San Diego, CA, 2009; ACM, New York, 2009; Vol. *39*, pp 1−11.
(4) Kikugawa, G.; Apostolov, R.; Kamiya, N.; Taiji, M.; Himeno, R.; Nakamura, H.; Yonezawa, Y. *J. Comput. Chem.* **2009**, *30*, 110−118.
(5) Freddolino, P. L.; Arkhipov, A. S.; Larson, S. B.; McPherson, A.; Schulten, K. *Structure* **2006**, *14*, 437−449.
(6) Day, R.; Paschek, D.; Garcia, A. E. *Proteins* **2010**, *78*, 1889−1899.
(7) Mittal, J.; Best, R. B. *Biophys. J.* **2010**, *99*, 26−28.
(8) Voelz, V. A.; Bowman, G. R.; Beauchamp, K.; Pande, V. S. *J. Am. Chem. Soc.* **2010**, *132*, 1526−1528.
(9) Pérez, A.; Luque, F. J.; Orozco, M. *J. Am. Chem. Soc.* **2007**, *129*, 14739−14745.
(10) Maragakis, P.; Lindorff-Larsen, K.; Eastwood, M. P.; Dror, R. O.; Klepeis, J. L.; Arkin, I. T.; Jensen, M. Ø.; Xu, H.; Trbovic, N.; Friesner, R. A.; Iii, A. G. P.; Shaw, D. E. *J. Phys. Chem. B* **2008**, *112*, 6155−6158.
(11) Dror, R. O.; Arlow, D. H.; Borhani, D. W.; Jensen, M. Ø.; Piana, S.; Shaw, D. E. *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *106*, 4689−4694.
(12) Bonvin, A. M. J. J.; Rosato, A.; Wassenaar, T. A. *J. Struct. Funct. Genomics* **2010**, *11*, 1−8.
(13) Hess, B.; Kutzner, C.; Van Der Spoel, D.; Lindahl, E. *J. Chem. Theory Comput.* **2008**, *4*, 435−447.
(14) Berendsen, H.; Van der Spoel, D.; Van Drunen, R. *Comput. Phys. Commun.* **1995**, *91*, 43−56.
(15) Wassenaar, T. A.; van Dijk, M.; Loureiro-Fereira, N.; van der Schot, G.; de Vries, S. J.; Schmitz, C. P. F.; van der Zwan, J.; Boelens, R.; Bonvin, A. M. J. J. WeNMR: Structural Biology on the Grid. *CEUR Workshop Proceedings* 2011, *819* (4), pp 1−8.
(16) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(17) de Vries, S. J.; van Dijk, M.; Bonvin, A. M. J. J. *Nat. Protoc.* **2010**, *5*, 883−897.
(18) Wassenaar, T. A.; Mark, A. E. *J. Comput. Chem.* **2006**, *27*, 316−325.
(19) Kaminski, G.; Friesner, R.; Tirado-Rives, J.; Jorgensen, W. *J. Phys. Chem. B* **2001**, *105*, 6474−6487.
(20) Tironi, I. G.; Sperb, R.; Smith, P. E.; van Gunsteren, W. F. *J. Chem. Phys.* **1995**, *102*, 5451−5459.
(21) Scott, W.; Hunenberger, P.; Tironi, I.; Mark, A.; Billeter, S.; Fennen, J.; Torda, A.; Huber, T.; Kruger, P.; van Gunsteren, W. *J. Phys. Chem. A* **1999**, *103*, 3596−3607.
(22) Schuler, L.; Daura, X.; van Gunsteren, W. *J. Comput. Chem.* **2001**, *22*, 1205−1218.
(23) Ewald, P. P. *Ann. Phys.* **1921**, 253−287.
(24) Cerutti, D. S.; Duke, R. E.; Darden, T. A.; Lybrand, T. P. *J. Chem. Theory Comput.* **2009**, *5*, 2322−2338.
(25) Bekker, H.; van den Berg, J. P.; Wassenaar, T. A. *J. Comput. Chem.* **2004**, *25*, 1037−1046.
(26) Amadei, A.; Chillemi, G.; Ceruso, M. A.; Grottesi, A.; Di Nola, A. *J. Chem. Phys.* **2000**, *112*, 9−23.
(27) Toukan, K.; Rahman, A. *Phys. Rev. B* **1985**, *31*, 2643−2648.
(28) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. *J. Comput. Chem.* **2005**, *26*, 1668−1688.
(29) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D. *J. Chem. Phys.* **1983**, *79*, 926−935.
(30) MacKerell, A. D.; Banavali, N.; Foloppe, N. *Biopolymers* **2000**, *56*, 257−265.
(31) Jorgensen, W.; Maxwell, D.; TiradoRives, J. *J. Am. Chem. Soc.* **1996**, *118*, 11225−11236.

(32) Sticht, H.; Bayer, P.; Willbold, D.; Dames, S.; Hilbich, C.; Beyreuther, K.; Frank, R. W.; Rösch, P. *Eur. J. Biochem.* **1995**, *233*, 293−298.
(33) Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. *Biochemistry* **1994**, *33*, 4721−4729.
(34) Wlodawer, A.; Nachman, J.; Gilliland, G. L.; Gallagher, W.; Woodward, C. *J. Mol. Biol.* **1987**, *198*, 469−480.
(35) Darden, T.; York, D.; Pedersen, L. *J. Chem. Phys.* **1993**, *98*, 10089−10092.
(36) Berendsen, H.; Postma, J.; van Gunsteren, W.; DiNola, A.; Haak, J. *J. Chem. Phys.* **1984**, *81*, 3684−3690.
(37) Bussi, G.; Donadio, D.; Parrinello, M. *J. Chem. Phys.* **2007**, *126*, 1−8.
(38) Berendsen, H. J. C.; van Gunsteren, W. F. Molecular dynamics simulations: Techniques and approaches. In *Molecular Liquids: Dynamics and Interactions*; Barnes, A. J., Orville-Thomas, W. J., Yarwood, J., Eds.; Reidel: Dorderecht, The Netherlands, 1984; pp 475−500.
(39) van Aalten, D. M.; Bywater, R.; Findlay, J. B.; Hendlich, M.; Hooft, R. W.; Vriend, G. *J. Comput.-Aided Mol. Des.* **1996**, *10*, 255−262.
(40) Malde, A. K.; Zuo, L.; Breeze, M.; Stroet, M.; Poger, D.; Nair, P. C.; Oostenbrink, C.; Mark, A. E. *J. Chem. Theory Comput.* **2011**, *7*, 4026−4037.
(41) Zoete, V.; Cuendet, M. A.; Grosdidier, A.; Michielin, O. *J. Comput. Chem.* **2011**, *32*, 2359−2368.
(42) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. *J. Mol. Graphics Modell.* **2006**, *25*, 247−260.
(43) Hospital, A.; Andrio, P.; Fenollosa, C.; Cicin-Sain, D.; Orozco, M.; Gelpi, J. L. *Bioinformatics* **2012**, advanced online publication.