JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

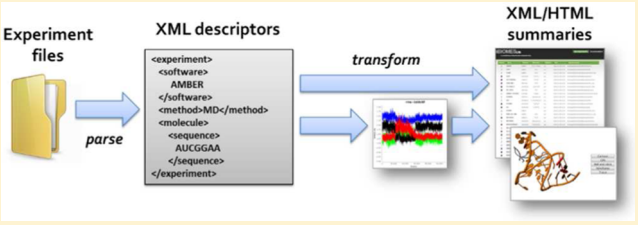
iBIOMES Lite: Summarizing Biomolecular Simulation Data in Limited Settings

Julien C. Thibault,[†] Thomas E. Cheatham, III,*[,‡] and Julio C. Facelli[†]

[†]Department of Biomedical Informatics and [‡]Department of Medicinal Chemistry, University of Utah, Salt Lake City, Utah 84112, United States

Supporting Information

ABSTRACT: As the amount of data generated by biomolecular simulations dramatically increases, new tools need to be developed to help manage this data at the individual investigator or small research group level. In this paper, we introduce iBIOMES Lite, a lightweight tool for biomolecular simulation data indexing and summarization. The main goal of iBIOMES Lite is to provide a simple interface to summarize computational experiments in a setting where the user might have limited privileges and limited access to IT resources. A command-line interface allows the user to summarize, publish, and search local simulation data sets. Published data sets are accessible via static hypertext markup language (HTML) pages that summarize the simulation protocols and also display data analysis graphically. The publication process is customized via extensible markup language (XML) descriptors while the HTML summary template is customized through extensible stylesheet language (XSL). iBIOMES Lite was tested on different platforms and at several national computing centers using various data sets generated through classical and quantum molecular dynamics, quantum chemistry, and QM/MM. The associated parsers currently support AMBER, GROMACS, Gaussian, and NWChem data set publication. The code is available at https://github.com/jcvthibault/ibiomes.



BACKGROUND

Over the past few decades high-performance computing resources have enabled the larger simulation community to push the limits of biomolecular simulations. As more computational power becomes available, researchers can tackle larger systems and simulate for longer time scales. While it was common practice to run the simulations on remote clusters and bring back the resulting data to the home institution, this paradigm is now beginning to break down. Data has to be postprocessed directly at the source to minimize data movements and minimize the amount of disk space necessary for storage. For example trajectories can be compressed and/or stripped of unnecessary information (e.g., solvent) before being copied over. Another approach is to simply run the analysis remotely, where the data resides. No matter which approach is preferred, researchers need to deal with a huge amount of data distributed over local and national resources. For example, molecular dynamics simulations of large biomolecules with thousands of atoms can easily generate terabytes of data on the microsecond time scale. Investigators need to store and catalog these data sets generated by students and collaborators for multiple years to comply with policies of the funding agencies, to ensure reproducibility of the data, and to comply with requests to see the data by other researchers.

Several repository architectures have been proposed to manage large biomolecular simulation data sets in a distributed environment. BioSimGrid[1] was deployed in the UK to integrate several computational centers into a grid, where data could be deposited, searched, and analyzed. Trajectory and provenance metadata were stored in a relational database. iBIOMES[2] on the other hand offers a distributed infrastructure that allows biomolecular simulation data indexing with data deposit (explicit copy) or in-place registration to avoid data movements. Trajectory files are stored and indexed via the iRODS distributed file system,[3] where metadata is represented as attribute−value−unit triplets. While these approaches might work well to manage large distributed environments, the deployment of such infrastructure depends on access to substantial IT expertise and resources, such as Web servers, relational databases, and distributed file systems, which may not be available to many single investigators or small research groups. Many researchers also depend on local or national computational and storage resources that are allocated for a finite period of time. Usage of these resources is usually very restrictive for security reasons, and the installation of heavy components such as databases is not an option to manage the data hosted at these remote locations. Another limitation of current repositories is the need to copy the simulation data to a remote server for publication. This can be a tedious task that requires extra storage cost if a copy of the data has to be kept at its original location. In this paper we introduce iBIOMES Lite, a new tool for biomolecular simulation data indexing and summarization, designed to run in limited settings, where the

Received: March 17, 2014

   dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Method	Name	Software	Molecules	Publisher	Date	Experiment path
	RNAMOD_DRD	AMBER	*RNA*	juji	2014-02-12 14:39	/home/juji/ibiomes/test/amber/rnamod_drd
	TUTORIAL3	AMBER	*Protein*	juji	2014-02-12 14:39	/home/juji/ibiomes/test/amber/tutorial3
	A-DNA	AMBER	*DNA*	juji	2014-02-12 14:39	/home/juji/ibiomes/test/amber/tutorial1/a-dna
	AM1-NUCLEOSIDE	AMBER	*DNA*	juji	2014-02-12 14:40	/home/juji/ibiomes/test/amber/sebomd/am1-nucleosi...
	MNDO-METHIONINE	AMBER	*Protein*	juji	2014-02-12 14:40	/home/juji/ibiomes/test/amber/sebomd/mndo-methion...
	PM3-ALANINEDIPEPTIDE	AMBER	*Protein*	juji	2014-02-12 14:40	/home/juji/ibiomes/test/amber/sebomd/pm3-alanined...
	ALADIP-QMMM-MD	AMBER	*Protein*	juji	2014-02-12 14:40	/home/juji/ibiomes/test/amber/qm_mm/aladip-qmmm-md
	REMD	AMBER	*Nucleic acid*	juji	2014-02-12 14:49	/home/juji/Documents/remd
	S3	GAMESS	S_3	juji	2014-02-12 14:41	/home/juji/ibiomes/test/gamess/S3
	ACETONITRILE	GAMESS	C_2H_3N	juji	2014-02-12 14:41	/home/juji/ibiomes/test/gamess/acetonitrile
	FOSFINA	GAUSSIAN	$C_{18}H_{15}P$	juji	2014-02-12 14:40	/home/juji/ibiomes/test/gaussian/fosfina
	TAMOXIFEN	GAUSSIAN	$C_{26}H_{29}NO_2$	juji	2014-02-12 14:40	/home/juji/ibiomes/test/gaussian/tamoxifen
	ACAC	GAUSSIAN	$C_5H_7O_2$	juji	2014-02-12 14:40	/home/juji/ibiomes/test/gaussian/acac
	FAD	GROMACS	$C_{27}H_{10}N_9O_{15}P_2$	juji	2014-02-12 14:40	/home/juji/ibiomes/test/gromacs/FAD
	QMMM_PROTEIN	GROMACS	*Protein* / $C_{18}H_4NO_6S$	juji	2014-02-12 14:40	/home/juji/ibiomes/test/gromacs/qmmm_protein
	SPEPTIDE	GROMACS	*Protein*	juji	2014-02-12 14:40	/home/juji/ibiomes/test/gromacs/speptide
	NAMD-AMBER	NAMD	*Protein* / *DNA*	juji	2014-02-12 14:40	/home/juji/ibiomes/test/namd/namd-amber
	DFT_BSSE	NWChem	H_2O	juji	2014-02-12 14:40	/home/juji/ibiomes/test/nwchem/dft_bsse
	PROP_CH3F	NWChem	CH_3F	juji	2014-02-12 14:40	/home/juji/ibiomes/test/nwchem/prop_ch3f
	N2_CCSD	NWChem	N_2	juji	2014-02-12 14:41	/home/juji/ibiomes/test/nwchem/n2_ccsd

Figure 1. Listing of published experiments in iBIOMES Lite Web site.

users might have limited privileges and limited access to IT resources. A command-line interface allows the user to summarize, publish, and search simulation data sets locally or remotely via secure shell (SSH). Published data sets are summarized through a static Web interface that describes the simulation protocols and graphically represent analysis results. iBIOMES Lite can be easily installed on any data server to enable summarizations of old data sets and figure out what their content is and what methods were used, or to facilitate progress tracking by exposing current simulation results. In contrast with simple tools such as Bookshelf[4] and UMM-MoDEL[5] that have been proposed to publish simulation data, but exhibit dependencies on database components, iBIOMES Lite allows data indexing and summarization while removing dependencies on external components that would require root access or special support for deployment.

DESIGN

Scope and Requirements. iBIOMES Lite's goal is to provide the means for individual researchers or teams to index and summarize their simulation data in limited settings, so they can keep track of their lab work and share progress or results with collaborators. The main user action supported by iBIOMES Lite is the publication of experiments: the user specifies a file directory or subdirectory that contains all the simulation files (input and output data), then with minimal input from the user, the tool generates a detailed description of the computational experiment workflow along with textual and graphical summaries, rendered through a simple Web interface. Once an experiment is published, it can be searched via keywords representing the experiment metadata (e.g., molecule name, residue sequence, computational method). Unlike the full-fledged iBIOMES repository,[2] iBIOMES Lite does not provide direct access to the files associated with the published experiments. All files are categorized and listed, but only files presenting analysis data are made available for download. This limitation was required to keep simplicity as a key design criterion for this tool. This criterion was applied at three different levels—deployment, usage, and customization—as follows:

Deployment. The tool should be able to run in most environments, independently from the operating system running on the host (e.g., Unix, Windows). The tool should also be able to run whether a graphical user interface is available or not. Root permissions should not be a prerequisite to install the program. This can be achieved by removing dependencies on heavy-weight components such as databases, Web servers, or specific file systems.

Usage. The tool should be usable in a multiuser and distributed environment by providing simple commands. The command-line interface provides a Unix-like interface to summarize simulation data, publish them into a static hypertext markup language (HTML) Web site, and perform keyword searches.

Customization. The publication process should be easily customizable by the user so that the resulting summaries provide an accurate and pertinent representation of the raw data. The actual code should not have to be modified to perform such customization. Instead customization should be enabled through templates and configuration files.

Web Interface. The entry point for the Web interface is a page listing all the published experiments, as shown in the iBIOMES Lite demonstration instance presented in Figure 1.

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

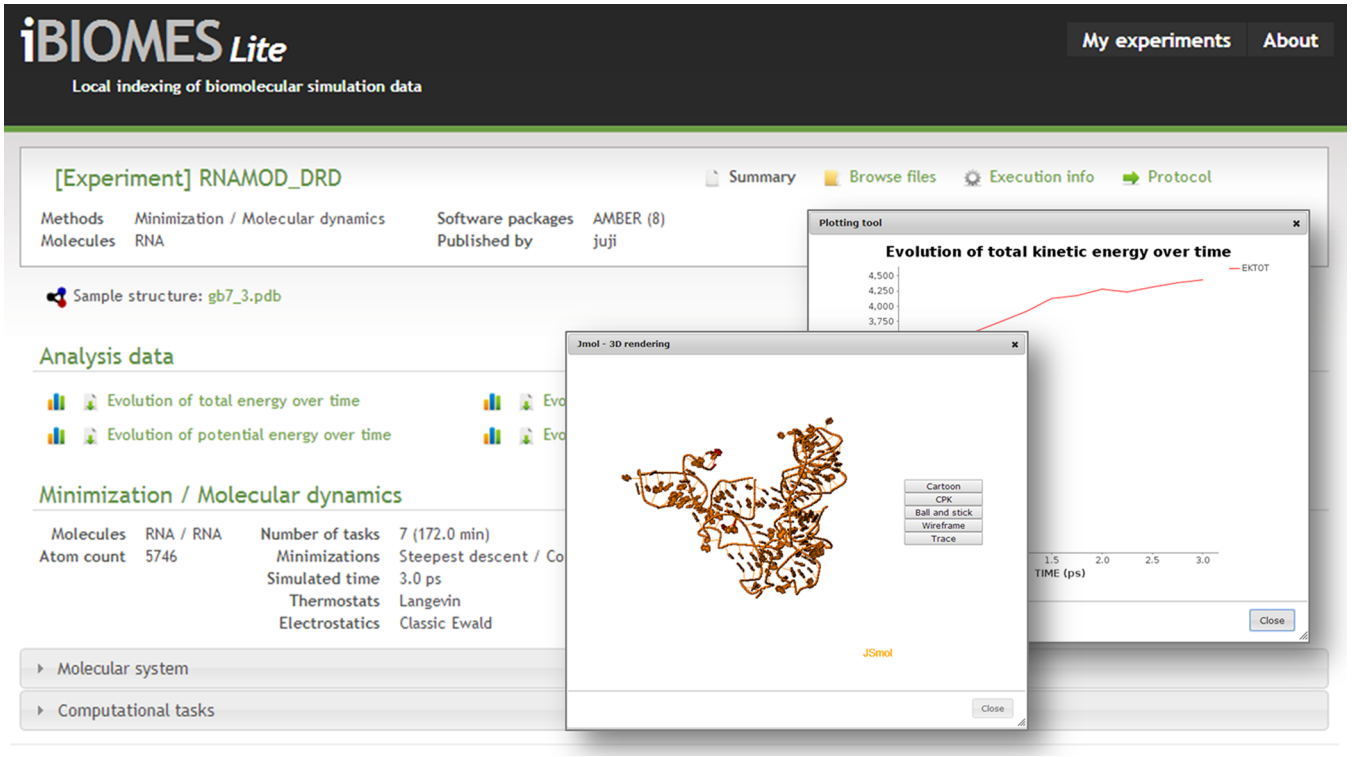


Figure 2. Example of an experiment summary page within the iBIOMES Lite Web site. In this example the page summarizes an MD simulation of RNA and enables graphical display via Jmol for 3D structure rendering and via autogenerated plots to present analysis data.

General information about the experiments (e.g., method, targeted molecular system, software package) is provided and can be used to sort the listing. By selecting one of the listed experiments, the user can access more details. Currently, each experiment is associated with four different HTML pages. The summary page (Figure 2) presents a summary of the experiment protocol along with possible analysis data, plots, and 3D structures, rendered via Jmol.[6] A second HTML page provides a tree view of the protocol used in the experiment, so that the user can access the details of interest, while keeping the overall picture of the workflow (Figure 3). A third HTML page provides a tree view that allows the user to browse the directory and subdirectories associated with the experiment and list their content (Figure 4). Finally a last HTML page gives details about the execution of the tasks and the computing environment (Figure 5). Execution times and resources used to run the tasks (e.g., number of CPUs and GPUs) are reported, along with hardware information (e.g., GPU architecture). Tasks that did not terminate correctly are flagged. This view is intended for users to track the progress of current simulations and assess the performance of their simulation engine within the host environment.

IMPLEMENTATION

Overview. iBIOMES Lite was implemented in Java 7 to ease the development of a platform-independent tool. Although Java 6 is arguably a more popular version, Java 7 offers enhanced file I/O libraries (NIO 2) that might prove to be useful for future developments (e.g., file change listeners, file tree searches), and it is still available at most US computing centers. A set of Bash scripts for Unix-like operating systems (i.e., Linux and Mac OS-X) and Win32 (.bat) scripts for Windows were written to wrap the Java calls into simple commands. These scripts can be easily called in a console locally or remotely, via SSH for example.

Publication Process. Users publish computational experiments to iBIOMES Lite to create HTML summaries and index their data for searches. A user publishes a computational experiment by specifying a directory or sub directory that contains all the simulation files (input and output) and the name of the software package that was used to generate these files (Figure 6). A set of file parsers extract topology, method, and parameter information to generate a representation of the simulation workflow, based on the data model introduced in previous work.[7] The workflow and file tree structures are stored as XML files then transformed into several HTML pages via XSL (extensible stylesheet language[8]). Plots are generated for analysis files when applicable then stored in the iBIOMES Lite Web directory along with the HTML files. Once an experiment has been published the information in the iBIOMES Lite Web folder can be updated by rerunning the publication command on the input experiment directory.

The final output of the publication process is a set of XML files, static HTML files, images, and other analysis data file (e.g., spreadsheet). These files can be exposed via an HTTP server such as Apache (http://httpd.apache.org/) or viewed locally if a graphical user interface is available. If neither option is available, the files can also be copied to a different host for rendering. Since the HTML is not generated on-the-fly by server-side code the Web content can always be copied without information loss. In the next sections we describe in more details the data extraction step performed by the file parsers and the data transformation step used to generate the HTML summaries.

Parsers. *Overview.* The role of the parsers is to map a given computational experiment file tree on disk to a logical representation of the protocol and output of the experiment.

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

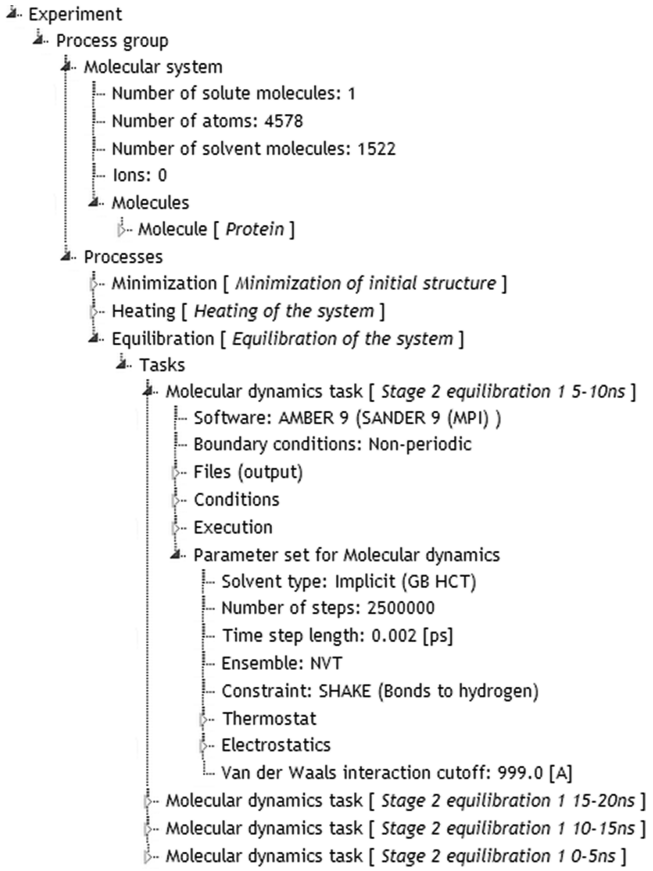


Figure 3. Experiment workflow example within the iBIOMES Lite Web site. In this example the experiment simulates a solvated protein and includes three processes: minimization, heating, and equilibration. Each process can include multiple tasks (or runs), which have method-specific parameters.

The data model introduced in our previous work[7] was used to guide the logical representation, for both the definition of the Java classes and the XML schema used to represent individual computational experiments, i.e. the simulations. The parsers work at the file level, extracting important data or metadata for file summary, and at the file tree level, trying to build the logical model based on the file directory structure and the file-extracted data.

The parsers can be configured through two different configuration files. An XML rule file can be used during the publication process to define which parsers should be used for which files and to define the way data is presented in the HTML summaries. A more general configuration file is used to specify default parameters for the publication process (e.g., path to default XML rule file, default software package context, console output options). Examples of these configuration files are provided in the Supporting Information.

File Parsers. The file parsers are format-specific, although they are expected to build certain common objects based on their type: topology, parameter/method, or hybrid. For example both the AMBER parameter/topology and CHARMM Protein Structure File (PSF) parsers are expected to build an object representing a molecular system, composed of one or multiple molecules, each represented by residues and/or atoms. On the other hand the AMBER MD input and NAMD configuration file parsers are building objects representing the methods and parameters used to run a computational task. Implementation of the parsers then requires understanding of the target format and the expected object(s) to build. All parsers target the data model introduced in previous work[7] to provide a common representation of the computational protocol that is not software-specific. The list of current parsers provides different levels of support for various software packages, including AMBER,[9] GROMACS,[10] NAMD,[11] NWChem,[12] and Gaussian.[13]

File Tree Parsers. The implementation of file tree parsers is not as straightforward since unlike the structure of a file which can be inferred from its format, the structure of a directory does not follow any strict rule. While we cannot force users to store their files following a given directory structure, manual inspection of files structure from many computational experiments performed in our lab by numerous graduate students and post docs lead us to assume that the protocol of the computational experiment can be inferred by parsing certain files if the original owner can provide a description of the file tree structure and the naming conventions they used to organize the data. Examples of such naming conventions include mappings between file format and suffix (e.g., ".trj" for a trajectory file or ".out" for an MD output/log file).

The preprocessing step in the mapping process is to parse all the files in the input directory and its subdirectories using the file-specific parsers. The resulting file tree associates each file with a set of descriptive data about the molecular system or computational methods. The second step is to build a logical representation of the computational experiment protocol using these objects. When publishing a new experiment the user needs to specify the main software package that was used to run the simulations (e.g., AMBER, NAMD, Gaussian, NWChem). Depending on this argument different rules are used to build the logical representation of the experiment. For example in AMBER, both MD input and MD output files can be used to retrieve the methods and parameters of a run. For most of the software packages the output/log files are preferred over input files to extract this type of data. Output files are typically richer as they usually repeat information from the input file(s) and provide explicit values to parameters that have not been set in the input, but which are used as the default values in the particular software. Output files can also present some calculation details, such as the evolution of the energy of system over a certain cycle of iterations, that can be easily exposed and of potential value to better understand the experiment protocol.

Other rules can be triggered based on the computational method used or the type of calculation performed. For example if minimization tasks and MD tasks are detected within the experiment, minimization tasks are grouped together, while MD tasks are divided into a "heating" process, an "equilibration" process and a "production MD" process. Heating tasks represent MD runs where temperature of the system is slowly increased, to eventually reach a reference temperature for the production runs. Distinction between equilibration and production runs is currently made based on the textual description of the task if it is available. Regular expressions were created to detect keywords such as "production", "prod", "equilibration", and "equil".

For replica-exchange MD (REMD), some extra step might be needed to group replicas for the same run together. In AMBER for example, an output file is created for each replica. In our data model, all replicas for a single run are grouped together under a single REMD task instead of having separate MD tasks representing individual replicas. Each REMD task is described like any other MD task and it also has a certain

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

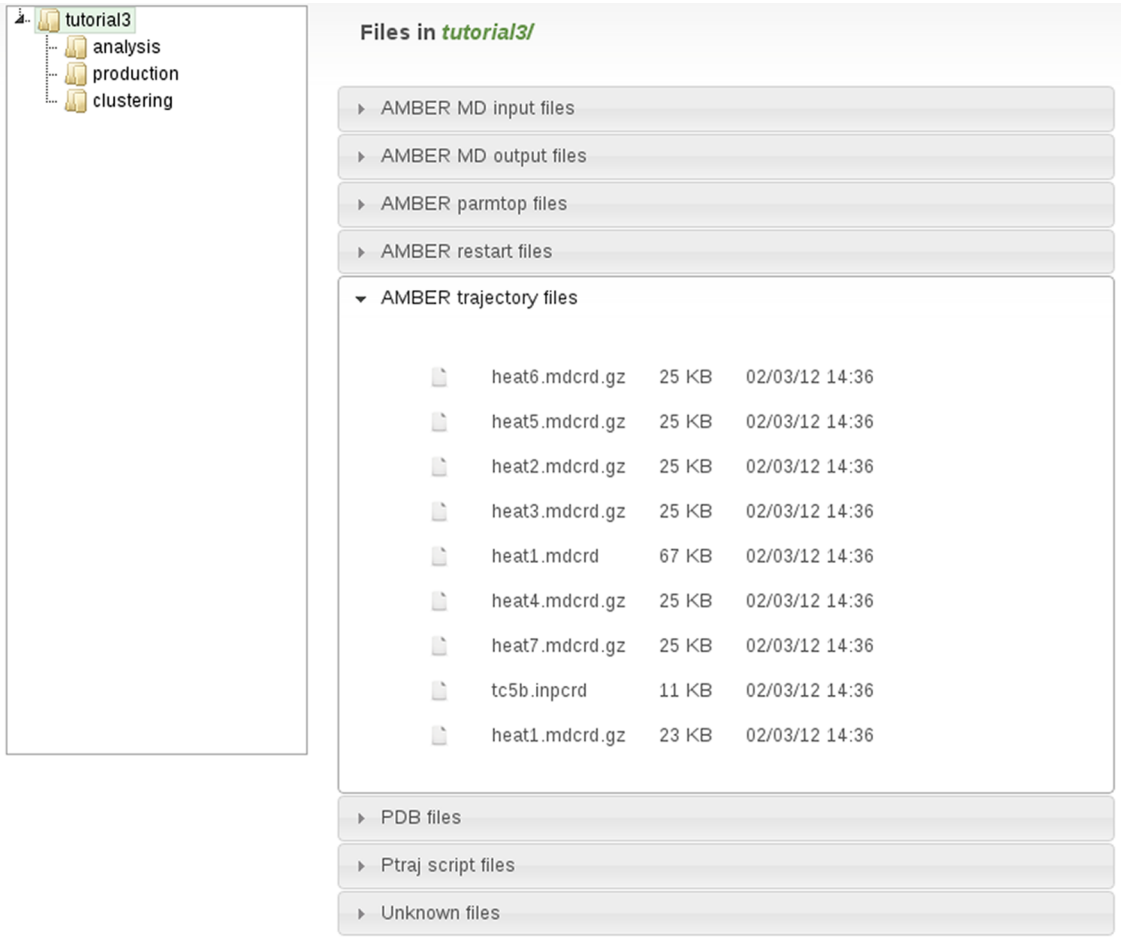


Figure 4. Experiment file listing within the iBIOMES Lite Web site. The tree view in the left panel enables directory browsing while the main panel lists the files in the selected directory, grouped by file format.

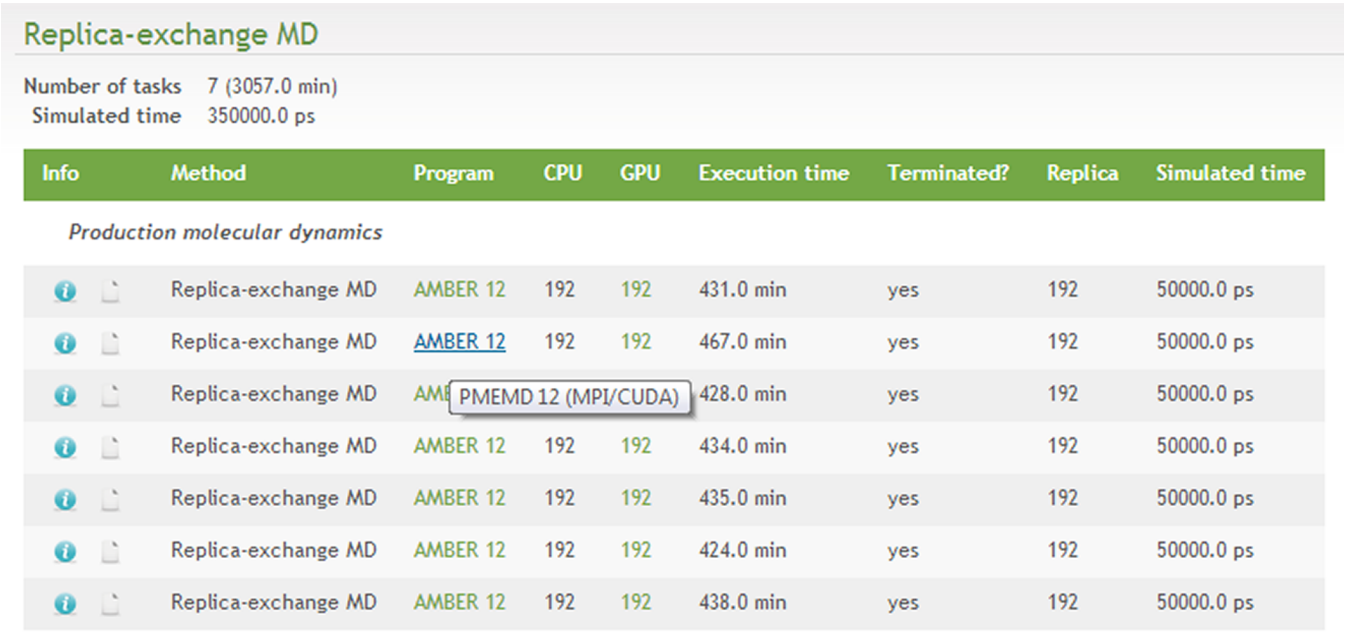


Figure 5. Execution summary within the iBIOMES Lite Web site. In this example, a list of REMD runs (192 replicas each) is presented to the user with job configuration details (e.g., number of CPUs and GPUs). Extra computing environment information, such as executable details and CPU/GPU architecture, can be displayed by hovering over the associated elements.

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

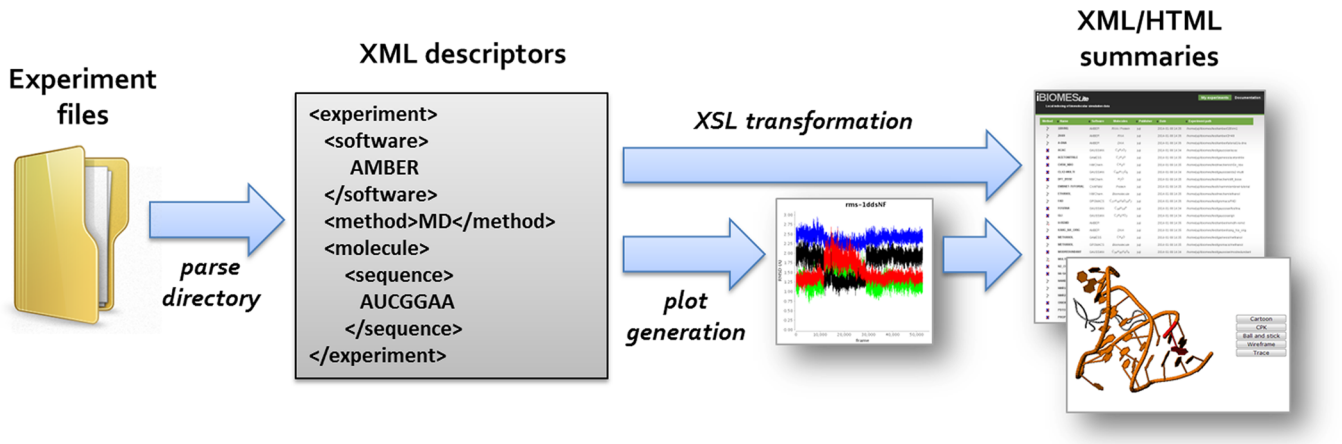


Figure 6. iBIOMES Lite publication process.

number of replicas and a type of exchange (e.g., temperature, Hamiltonian, multidimensional). This representation helps summarizing the data, especially when running REMD simulations with hundreds of replicas. By default REMD output files stored in the same folder are assumed to represent replicas from the same group. This would apply for example if a user stored three four-replica REMD runs in three different folders, each with four output files. Experience shows that this approach is not unique, and some people might prefer to have all REMD output in a single folder. Replica identification and grouping is then based on file naming conventions. Using the same example, a user could store all the REMD output files in a single folder and name the files using the pattern that identifies both the run and the replica within this run, such as

$$\texttt{remd.[ID}_{\text{RUN}}\texttt{].[ID}_{\text{REPLICA}}\texttt{].out,}$$

$$\text{where } 0 \le \text{ID}_{\text{RUN}} \le 2 \text{ and } 0 \le \text{ID}_{\text{REPLICA}} \le 3.$$

The user can specify this type of naming convention in the iBIOMES Lite general configuration file or at run time using the *-remd* command line argument. If no run identifier is present in the name pattern then grouping is solely based on the directory structure. This type of rule-based grouping is currently applied to REMD tasks only but it could be expended to include any type of parallel enhanced sampling task.

Data Transformations. *XML Representation.* After the logical model of an experiment is built within the Java code it is stored on disk as an XML file. Mapping between the Java object-oriented data model and the XML schema is performed via JAXB (Java Architecture for XML Binding). An example of such XML is presented in the Supporting Information. A second XML file is generated based on the file tree structure, where each file is associated with a set of metadata, represented as attribute-value-units (AVU) triplets. This representation is very similar to the approach used for the iBIOMES repository[2] to enable indexing within iRODS (Integrated Rule-Oriented Data System[3]). An example of such XML file tree is given in the Supporting Information. The AVUs are derived from the objects extracted by the file parsers, such as molecular system definitions or parameter sets. Each of these entities implement a *getMetadata()* method that translates the logical entity (object) into a list of AVUs. For example the *getMetadata()* method for the *Thermostat* class will generate AVUs for the followings attributes: THERMOSTAT_ALGORITHM (e.g., Berendsen, Langevin) and THERMOSTAT_TIME_CONSTANT if applicable (e.g., 2 ps for a Berendsen thermostat).

These XML documents provide two different perspectives on the data: one that emphasizes on the experimental protocol, or effectively the logical view, and another one that emphasizes on the physical organization of the input and output files. While the first view can provide some insight on the protocol used to run the simulations, the second view enables simple data indexing via keywords. A copy of these XML files is stored directly in the experiment folder. Another copy is pushed to the iBIOMES Lite Web folder, in a subdirectory dedicated to the experiment. A separate XML document representing the list of published experiments is also updated by copying experiment-level AVUs from the XML document storing the experiment file tree.

Analysis Data. Beside the experimental protocol and the file tree, iBIOMES Lite can present analysis data in the experiment summary page. The user can edit an XML configuration file to define which piece of data should be presented and how it should be presented (see example in the Supporting Information). This is achieved by associating file name patterns to analysis descriptions, as introduced in iBIOMES.[2] Any file that is marked as analysis data is copied to the iBIOMES Lite Web folder to enable display and/or download. For example PDB files that are marked as analysis data can be rendered via Jmol,[6] and image files (e.g., PNG, JPEG) are presented as thumbnails linking to a copy of the original picture. For column delimited text files (e.g., tab- or comma-delimited files) the tool attempts to create a graphical representation of the content. The XML configuration files can be used to define the type of plot to be generated (e.g., line plot, histogram, heatmap), its labels, units, and title (see example in the Supporting Information). The resulting plot is exported as an image and copied over to the iBIOMES Lite Web folder, along with the original data file.

Transformation. Once the XML files and data files have been copied to the iBIOMES Lite Web directory, all data and metadata of interest are ready to be visually rendered by transforming the XML into HTML. Multiple XSL stylesheets define the mappings between the XML and the various HTML pages necessary to list the published experiments and provide details about individual experiments. The actual XSL 2.0 based transformation process in the Java code is performed via the Saxon processor.[14] Since XSL stylesheets are defined as separate documents one could easily customize these HTML templates to fit their need.

Shared iBIOMES Lite Web Folder for Multiuser Use. iBIOMES Lite allows multiple users to share the same Web

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

directory to publish experiments. This means that all the members of a lab for example can publish experiments stored on a shared file system to a single portal. From a user-interface perspective, information about the publication event needs to be tracked: each experiment is associated with a publication date (different from the data set creation date) and a publisher (i.e., the file system username). From a publication perspective, safeguards have to be created to ensure data integrity when two users try to publish an experiment simultaneously. If both users try to publish the same experiment then one should be blocked to allow the other user's action to parse the associated directory and generate the descriptor files. Whether the target experiments are different or not, the Web directory containing the listing and the index of experiments should not be updated concurrently.

A locking system was implemented to prevent concurrent updates. If somehow two users are trying to publish the same experiment folder concurrently, the second user's publication action is automatically canceled and the user is warned. If two users are trying to publish different experiments simultaneously, updates from the second user on the experiment listing will be queued until the first users' publication process is over.

Commands. Various Unix-like commands are available to manage the published experiments in iBIOMES Lite. A complete description of these commands is available on the iBIOMES Wiki (http://ibiomes.chpc.utah.edu/mediawiki/). Here we only present a summary of the most important ones: the publish (ibiomes-lite-publish), search (ibiomes-lite-search), and clean (ibiomes-lite-clean) commands.

Publish Experiments. To publish an experiment into iBIOMES Lite—i.e. to parse the experiment folder and generate the associated Web content—or to update the Web content for a given experiment, one should use the following command:

```
ibiomes-lite-publish -i <experiment-dir> [-s software] [-x xml-descriptor]
[...]
[experiment-dir] Path to the root of the experiment directory
[software] Name of the software package used to run the
simulation/calculations (e.g. amber, nwchem)
[xml-descriptor] Path to the XML descriptor that specifies metadata
generation rules. If no file is specified default values defined in the API
are used.
```

Search Experiments. iBIOMES Lite offers a simple search function: the user provides a list of keywords that are matched against the AVU values in the XML document listing all the published experiments. Paths to experiments that contain all provided keywords are returned. Searches are performed via the *ibiomes-lite-search* command, defined as

```
ibiomes-lite-search < keywords >

[keywords] List of keywords separated by '+' character. Wildcards can be
specified using '%'. Example:

ibiomes-lite-search %dynamics+rna+amber.
2 experiment(s) found:
      [0] /home/user1/ibiomes/test/amber/rnamodrd
      [1] /home/user1/ibiomes/test/amber/tutorial1
```

Clean Web Content. Remove content (XML and HTML) from iBIOMES Lite Web site. XML descriptors at the experiment directory level are conserved, and can be published again. If the -i option is not specified then all experiments are removed:

```
ibiomes-lite-clean

ibiomes-lite-clean -i < experiment-dir >

[experiment-dir] Physical path to the experiment to remove from iBIOMES Lite
```

TESTS IN LIMITED SETTINGS

Methods. A critical test for iBIOMES Lite is to demonstrate its ability to work in a variety of environments, including large computational clusters hosted by national centers and single Principal Investigator (PI) laboratories. A successful deployment here is defined by the following criteria:

1. All prerequisites (i.e., Java 7) are installed or can be installed on the targeted system.
2. The user can install iBIOMES Lite on the targeted system, i.e. copy the files and set up the necessary environment variables, and configuration parameters.
3. The user can publish data sets within the targeted system and visualize the generated Web site within this system or an external one (e.g., home institution).

To demonstrate these capabilities iBIOMES Lite was deployed on various machines, such as desktop computers and laptops running different operating systems and at several US National Science Foundation funded computational centers.

Results. iBIOMES Lite was successfully deployed on different desktop computers and laptops, running the following operating systems: Linux (Fedora Core 18), Windows 7, and Mac OS X 10. iBIOMES Lite was also deployed at the following facilities: the Center for High Performance Computing (CHPC) at the University of Utah, the National Center for Supercomputing Applications (NCSA) Petascale Computing Facility, the Texas Advanced Computing Center (TACC), and the San Diego Supercomputing Center (SDSC). The actual computational environments targeted for testing purpose are described in Table 1.

Table 1. List of Computing Centers Where iBIOMES Lite Was Successfully Deployed

resource	center	description	OS	Java version
Blue Waters	NCSA	Cray XE6/XK7 system, 22500 CPU nodes and 4200 CPU/GPU nodes	UNICOS	1.7.0_07-b10
Stampede	TACC	6400 nodes, InfiniBand Mellanox Switches/ HCAs	BusyBox	1.7.0_45-b18
Gordon	SDSC	1024 nodes, QDR InfiniBand interconnect	CentOS	1.7.0_13-b20
Ember	CHPC	262 nodes, 3144 cores, InfiniBand and gigabit ethernet interconnects	RHEL 6.4	1.7.0_03-b04

More detailed benchmarking on the parser was performed on Blue Waters (NCSA) and Stampede (TACC). The data sets descriptions and associated directory parsing timings are reported in Table 2. All the reported timings were obtained by submitting several batch jobs to these two clusters, using a single computational node. The reported average and standard deviation (std dev) for the processing times were calculated based on 10 jobs for each data set.

Dependence between log file (AMBER MD output) sizes and parser execution times is presented in Figure 7. As expected, the larger the aggregated size of all log files the longer the execution time since MD output files are the main target of the parsers. The timings presented here are only presented as a rough estimate for various types of AMBER data sets. In our example data sets the number of topology files (e.g., PDB, AMBER parameter/topology) is fairly small compared to the number of MD output files but the timings are still dependent on these files. For example if a large number of PDB files

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 2. Parsers' Benchmarking on Blue Waters (NCSA) and Stampede (TACC)[a]

data set	1	2	3	4	5	6
resource	Blue Waters	Blue Waters	Blue Waters	Stampede	Stampede	Stampede
system description	RNA tetranucleotide[15]	RNA tetraloop	RNA tetraloop	polymer–ligand complex	coiled-coil dimer	protein
replicas/copies	192 REMD replicas	360 REMD replicas	576 REMD replicas	8 ligand configurations	5 config	1
number of atoms	7622	6071	15599	~122000	38744	~22500
number of runs	1	1	1	147/config	8/config	12
trajectory length	9600 ns	7200 ns	17280 ns	6960 ns	1000 ns	300 ns
number of files	1160	2536	4043	3425	357	404
total directory size	659 GB	54 GB	315 GB	816 GB	221 GB	24 GB
log write interval	2 ps	10 ps	2 ps	10 ps	2 ps	2 ps
average log file size	16 MB	1.8 MB	9.5 MB	0.5 MB	8 MB	20 MB
total processed size	3072 MB	648 MB	5472 MB	588 MB	320 MB	240 MB
Execution Time						
average (s)	264.2	119.4	504.6	64.8	26.4	14.9
std dev (s)	58.3	2.1	43.7	1.2	0.7	0.3

[a]The value reported for the "trajectory length" is the aggregated length of all trajectories in the input folder. The value reported as "total processed size" is the sum of the sizes of all the MD output files in the directory.

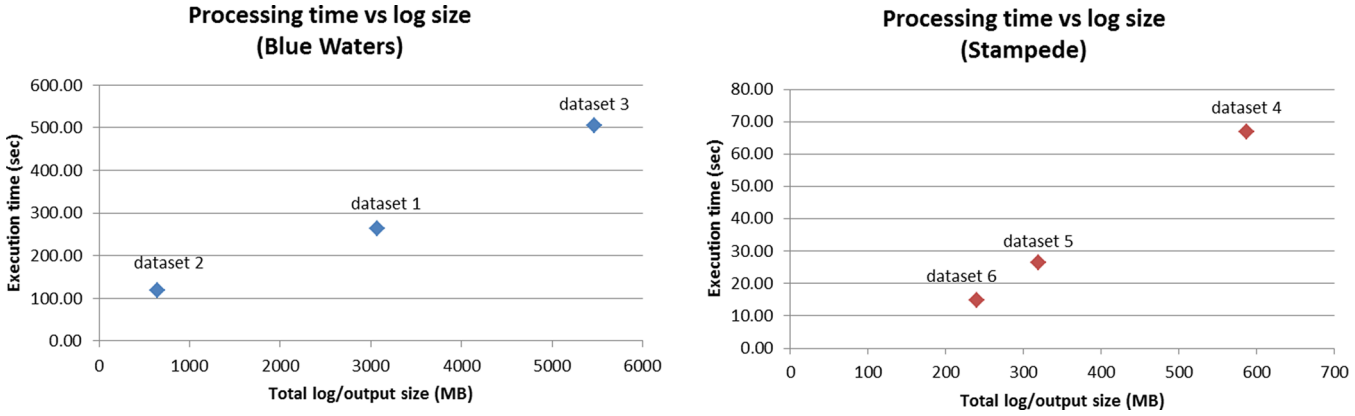


Figure 7. Dependence between parsing execution time and total output/log file size.

representing trajectory snapshots or representative structures with solvent information is present in the input directory, the MD output might not have as much impact on the overall parsers' performance. Note that trajectory files (e.g., AMBER NetCDF, CHARMM DCD) are not actually parsed since they are typically very large (~MB-TB), and they do not provide extra information about the topology or methods used in the simulation.

The parsers were also tested on Blue Waters using an interactive session. The parsers seem to be faster with an average execution time of 94.20 s, versus 119.4 s for the equivalent batch job. The standard deviation was higher (14.85 vs 2.1 s), which can be explained by the fact that the interactive node was shared with other users running various tasks.

DISCUSSION

Thanks to its simplicity, iBIOMES Lite can be deployed in limited environments where users have limited permissions and no access to heavy components such as database system managers. More importantly, we showed here that iBIOMES Lite can be used at major computational centers where Big Data is generated.

Summarization does not require bringing back the raw data to the home institution: iBIOMES Lite can be run at the source despite the limitations due to security concerns in such infrastructures. Since the published summaries are static and provide a compressed view of the simulation, the results of the publications can be easily copied to a new location for rendering via the Web or simply to centralize the summaries from different computing centers at a single location. Scripts could be created to automate this process, as well as to regenerate the summaries to make sure that they are up to date with the associated raw data. Since the publication process is performed via a command line interface, the iBIOMES Lite summarization step can be added to a regular simulation job description when running in a cluster. Another alternative when targeting data hosted at a computational center is to run the publication process via an interactive session. For very large data sets with thousands of files the parsers might take over half an hour to go through all the files. Running such tasks on the login nodes of a cluster is usually not recommended by the hosting institution as other users might observe a dramatic slowdown when trying to access their data or submit a job. Most computing centers allow users to request interactive sessions, which are usually provided within minutes, unlike batch job submissions which might stay queued for hours or days.

Data publication is done through a simple command-line interface that controls the software-specific parsers to build the summaries. The user only needs to specify the input folder and the software package that was used to generate the data to create summaries that can be exposed on the Web. The specification of the software package version is not necessary since the parsing process favors output/log files over input files to avoid the use of default values that might change between

dx.doi.org/10.1021/ci500173w | J. Chem. Inf. Model. XXXX, XXX, XXX–XXX

versions. Moreover, several parsers already include parsing functions for different format versions where keywords or structure might change (e.g., AMBER MD output, SDF file).

One of the current limitations of iBIOMES Lite is that only simple or common directory structures will provide an accurate representation of the simulation workflow. Since there is no input from the user regarding the file tree structure certain assumptions have to be made about the workflow and the directory structure to build a representation of the workflow. An example of constraint that is currently imposed at the workflow level is the sequence minimization−heating−equilibration−production that has to be followed by all MD experiments (with each component being optional). An example of directory structure assumption is for multicopy simulations, where each copy should be in a different subfolder. Therefore, our current parsers and protocol model builders will not be adapted to all types of experiments and directory structures, but this limitation should be circumvented in the future by including more configurable rules based on naming conventions, file content, computational methods, and textual descriptions to enable an accurate representation of the experiment protocol with minimal input from the user. Although most demonstrations for iBIOMES Lite have been done through the publication of AMBER-generated data sets, the parsers support data sets generated by other MD engines such as GROMACS and NAMD. The development of the data model and parsers has been guided by our experience with AMBER but the support for other software packages has allowed us to avoid software-specific data representations and parsing rules. Support for more software packages will be added as potential users show interest. Parsers for files associated with MD engines that use static configuration files (versus script-based input) can be implemented with minimal effort if the formats are well-defined and data is available to the developers for testing. Parsers for QM data sets (e.g., GAUSSIAN, NWChem) were also developed to demonstrate the generalizability of the data model and the Web interface. Although nowadays MD is a de facto standard approach to run biomolecular simulations, QM cannot be excluded from this realm. First MD can be dependent on QM when new force field parameters have to be created for nonstandard residues or small ligands. Then QM has promises in the study of biomolecules, at least for small systems.[16] Moreover, we note that we spent considerable initial effort on generating QM data parsers for the prototype iBIOMES repository implementation.[2] Unfortunately, due to serious limitations and restrictions on the release or publication of raw Gaussian output by Gaussian, Inc., open "publication" of Gaussian data in iBIOMES is not recommended. However, use of the functional parsers in iBIOMES Lite provides a simple way to summarize and track these data sets without having to provide access to the raw data.

The inclusion of less common or more complex methods in the data model such as replica-exchange MD, QM/MM, and quantum MD has proven the decomposition of parameters into sets of method-specific parameters to be fairly generalizable. These methods are currently supported only for the AMBER software package, which enables QM/MM MD,[17] semi-empirical Born−Oppenheimer MD (SEBOMD[18]), and multi-dimensional replica-exchange MD.[15] A current list of the file formats and computational methods supported by the parsers is available on the Wiki at http://ibiomes.chpc.utah.edu/mediawiki/index.php/IBIOMES_parsers. The initial rationale behind the development of iBIOMES Lite was the need for a simple tool that would be able to mimic the features offered by the iBIOMES repository[2] in a nondistributed environment controlled by a strict security policy. This has been a successful attempt as iBIOMES Lite can create rich summaries with graphical rendering (3D structures, data plots) and basic search capabilities. One advantage of iBIOMES Lite over the distributed repository is the ability to provide a detailed and logical description of the computational experiment protocol via XML transformation. The current AVU model used by the iBIOMES repository to index data is very flexible but relationships between data elements cannot be described. The addition of a relational database to the repository architecture to keep track of the experiment workflow is part of our effort to provide a generic infrastructure for biomolecular simulation data sharing.[7] One of the major limitations of iBIOMES Lite, by design, is the fact that the Web interface does not provide access to the raw data. iBIOMES Lite is not a replacement for data repositories. Instead it should be seen as a way for researchers to summarize data at the source for progress tracking and result sharing. Our end-goal is to enable the integration of iBIOMES Lite summaries into the iBIOMES repository. Researchers would be able to summarize their data within a computational center that does not support iRODS-based data transfers, and publish the summary into the iBIOMES repository. The raw data would not be available for download but users would be able to search for both full experiments data sets and experiment summaries via a single entry point: the repository Web portal. This effort is currently supported by a common data model, a common set of parsers, and similar Web interfaces.

CONCLUSION

iBIOMES Lite provides the means for researchers to track and share biomolecular simulation data sets via automatic summarization. Summaries are supported by a software-independent data model that can describe quantum chemistry, classical and quantum MD, REMD, and QM/MM data sets. Thanks to a simple design, the tool can be easily installed on machines where users have limited privileges, whether they are hosted locally or at a national computing center. iBIOMES Lite is an open-source project and is part of the iBIOMES distribution, available at https://github.com/jcvthibault/ibiomes.

ASSOCIATED CONTENT

Supporting Information

XML representation of the file tree associated with a computational experiment. XML representation of the computational experiment protocol. XML parser rule file example 1: defining file format and descriptions based on naming conventions. XML parser rule file example 2: identifying analysis data. XML parser rule file example 3: defining rules for plot generation. General parser configuration file example. This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author

*Mailing address: University of Utah, Department of Medicinal Chemistry, 30 South 2000 East, Room 307, Salt Lake City, UT 84112-5820. E-mail: tec3@utah.edu.

Funding
Research funding came from the NSF CHE-1266307 and NIH R01-GM081411.

Notes
The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

Computational support was provided by the Center for High Performance Computing (CHPC) at the University of Utah, the Blue Waters sustained-petascale computing project (NSF OCI 07-25070 and PRAC OCI-1036208), and the NSF Extreme Science and Engineering Discovery Environment (XSEDE, OCI-1053575) and allocation MCA01S027P. Thanks to the AMBER developer community who provided us with various data sets to test our parsers. We also would like to thank Christina Bergonzo, Rodrigo Galindo-Murillo, and Sean Cornillie for helping us benchmark iBIOMES Lite at several national computing centers.

■ ABBREVIATIONS

MD, molecular dynamics; QM, quantum mechanics; QM/MM, quantum mechanics/molecular mechanics; AVU, attribute−value−unit; HPC, high-performance computing

■ REFERENCES

(1) Ng, M. H.; Johnston, S.; Wu, B.; Murdock, S. E.; Tai, K.; Fangohr, H.; Cox, S. J.; Essex, J. W.; Sansom, M. S. P.; Jeffreys, P. BioSimGrid: Grid-Enabled Biomolecular Simulation Data Storage and Analysis. *Future Gener. Comp. Sy.* **2006**, *22*, 657−664.

(2) Thibault, J. C.; Facelli, J. C.; Cheatham, T. E., 3rd iBIOMES: Managing and Sharing Biomolecular Simulation Data in a Distributed Environment. *J. Chem. Inf. Model.* **2013**, *53*, 726−736.

(3) Rajasekar, A.; Moore, R.; Hou, C.; Lee, C. A.; Marciano, R.; de Torcy, A.; Wan, M.; Schroeder, W.; Chen, S. Y.; Gilbert, L. iRODS Primer: Integrated Rule-Oriented Data System. *Synth. Lect. Inf. Concepts Retrieval Services* **2010**, *2*, 1−143.

(4) Vohra, S.; Hall, B. A.; Holdbrook, D. A.; Khalid, S.; Biggin, P. C. Bookshelf: A Simple Curation System for the Storage of Biomolecular Simulation Data. *Database: J. Biol. Databases Curation* **2010**, DOI: 10.1093/database/baq033.

(5) Goni, R.; Apostolov, R.; Lundborg, M.; Bernau, C.; Jamitzky, F.; Laure, E.; Lindhal, E.; Andrio, P.; Becerra, Y.; Orozco, M.; Lluis Gelpi, J. *White Paper on Standards for data handling*; ScalaLife, 2013.

(6) Herráez, A. Biomolecules in the Computer: Jmol to the Rescue. *Biochem. Mol. Biol. Educ.* **2006**, *34*, 255−261.

(7) Thibault, J. C.; Roe, D. R.; Facelli, J. C.; Cheatham, T. E., III Data model, dictionaries, and desiderata for biomolecular simulation data indexing and sharing. *J. Cheminf.* **2014**, *6*, 4.

(8) W3C. The Extensible Stylesheet Language Family (XSL). http://www.w3.org/Style/XSL/ (last accessed 3/17/2014).

(9) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber Biomolecular Simulation Programs. *J. Comput. Chem.* **2005**, *26*, 1668−1688.

(10) Pronk, S.; Pall, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; Hess, B.; Lindahl, E. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* **2013**, *29*, 845−54.

(11) 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.

(12) Valiev, M.; Bylaska, E. J.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Van Dam, H. J. J.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T. L. NWChem: A Comprehensive and Scalable Open-Source Solution for Large Scale Molecular Simulations. *Comput. Phys. Commun.* **2010**, *181*, 1477−1489.

(13) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian 09*, Revision C. 01; Gaussian, Inc: Wallingford, CT, 2009.

(14) *Saxon-HE (Home Edition)*, 9.5; Saxonica, 2014.

(15) Bergonzo, C.; Henriksen, N. M.; Roe, D. R.; Swails, J. M.; Roitberg, A. E.; Cheatham, T. E., III Multidimensional Replica Exchange Molecular Dynamics Yields a Converged Ensemble of an RNA Tetranucleotide. *J. Chem. Theory Comput.* **2013**, *10*, 492−499.

(16) Šponer, J.; Šponer, J. E.; Mládek, A.; Banáš, P.; Jurečka, P.; Otyepka, M. How to understand quantum chemical computations on DNA and RNA systems? A practical guide for non-specialists. *Methods* **2013**, *64*, 3−11.

(17) Götz, A. W.; Clark, M. A.; Walker, R. C. An extensible interface for QM/MM molecular dynamics simulations with AMBER. *J. Comput. Chem.* **2014**, *35*, 95−108.

(18) Monard, G. SEBOMD (SemiEmpirical Born-Oppenheimer Molecular Dynamics): Techniques and applications. In *CECAM Workshop: Approximate Quantum-Methods: Advances, Challenges & Perspectives*, University of Bremen, Germany, 2010.