Published in final edited form as:

SAR QSAR Environ Res. 2015 ; 26(0): 577–593. doi:10.1080/1062936X.2015.1076515.

Development of an informatics infrastructure for data exchange of biomolecular simulations: architecture, data models and ontology^{\$}

J. C. Thibault¹, D. R. Roe², K. Eilbeck¹, T. E. Cheatham III², and J. C. Facelli^{1,*}

¹Department of Biomedical Informatics, University of Utah, Salt Lake City, Utah, US

²Department of Medicinal Chemistry and Center for High Performance Computing, University of Utah, Salt Lake City, Utah, US

Abstract

Biomolecular simulations aim to simulate structure, dynamics, interactions, and energetics of complex biomolecular systems. With the recent advances in hardware, it is now possible to use more complex and accurate models, but also reach time scales that are biologically significant. Molecular simulations have become a standard tool for toxicology and pharmacology research, but organizing and sharing data – both within the same organization and among different ones – remains a substantial challenge. In this paper we review our recent work leading to the development of a comprehensive informatics infrastructure to facilitate the organization and exchange of biomolecular simulations data. Our efforts include the design of data models and dictionary tools that allow the standardization of the metadata used to describe the biomedical simulations, the development of a thesaurus and ontology for computational reasoning when searching for biomolecular simulations in distributed environments, and the development of systems based on these models to manage and share the data at a large scale (iBIOMES), and within smaller groups of researchers at laboratory scale (iBIOMES Lite), that take advantage of the standardization of the meta data used to describe biomolecular simulations.

Keywords

molecular simulations; vocabularies; data repositories; ontologies; data sharing; distributed computing

Introduction

Biomolecular simulations aim to simulate structure, dynamics, interactions, and energetics of complex biomolecular systems. Understanding biological phenomena with these methods may facilitate the design of better drugs, therapies, catalysts and nanotechnology, increasing our understanding of their pharmacological and toxicological properties. With the recent

^{\$}Presented at the 8th International Symposium on Computational Methods in Toxicology and Pharmacology Integrating Internet Resources, CMTPI-2015, June 21-25, 2015, Chios, Greece.

*Corresponding author. Julio.Facelli@utah.edu.

advances in hardware, it is now possible to use more complex and accurate models, but also reach time scales that are biologically significant. For instance when simulating biomolecular dynamics on the microsecond scale, one can generate molecular dynamics trajectories that require TeraBytes (TB) of data on disk. While the computing power has dramatically increased, our ability to manage, store, analyze, and move large datasets is still limited. This creates new challenges for the community. Central repositories or even at the lab level are desirable to facilitate data management, analysis, and sharing, but they will require new methods to catalog existing datasets by keeping them in place or improved mechanisms for distributed data storage and movements.

Biomolecular simulations and computational chemistry are dominated by two classes of methods: Molecular Dynamics (MD) and Quantum Mechanics (QM). Many variations of these methods exist, along with hybrid approaches that combine them. A wide variety of MD and QM codes are available to the scientific community [4-13], making data sharing outside the owner's institution a substantial challenge, despite the potential scientific benefits.

Several projects have tried to tackle the problem of molecular simulation data storing and/or sharing. We can distinguish two types of infrastructure: one that is purely based on relational databases, and one that keeps references to the raw input and output files and only stores simulation metadata in a relational database. The BioSimGrid project [14] and the Dynameomics project [15] belong to this first category, where trajectory information is stored directly into database tables. Scalability of pure relational databases using this approach becomes problematic as we reach larger molecular systems and biologically-relevant time scales. Another issue for these databases is the lack of analysis tools as most current analysis tools perform their calculations on the raw files, and not on database tables. The eMinerals project [16-17], and the MoDEL[18-19], databases adopted a different approach where the raw output files are made available and searchable through a database that stores information about the runs (e.g. PDB ID, molecule name). Other example is the ABC (Ascona B-DNA Consortium) initiative, led by multiple laboratories distributed all over the world. A large series of Molecular Dynamics simulations of B-DNA were run to expose sequence-specific nucleic acid structure and dynamics [20-21].

In this paper, we review recent advances at the University of Utah to facilitate the exchange of biomolecular simulation data. The paper includes a summary discussion of the topics that have been previously reported in detail in the literature, including data models, dictionaries and the iBIOMES and iBIOMES Lite infrastructures, but also provides a great deal of detail on our work on ontologies, which has not been previously published.

Data model, dictionaries for biomolecular simulation data indexing and sharing

Perhaps one of the main barriers to build repositories for biomolecular simulations is the lack of standard models and vocabularies to represent these simulations. To the authors' knowledge before our work [22] no published study had tried to assess the needs of the community regarding biomolecular simulation repository data models and vocabularies. One

of our goals has been to identify which pieces of information are considered essential by researchers and how they should be organized in a computable manner, so that users can build structured queries to retrieve simulations or calculations of interest. In the aforementioned [22] study we formalized the need for data models and dictionaries that should not only provide a description of the various independent computational tasks performed but also a high-level description of the overall simulated experiment. We first used a survey to identify these common data elements that help to understand what was simulated, how, and in which context. These concepts were grouped into the following categories: authorship (author, grants and publications related to the simulated experiment), methods (computational method and associated parameters), molecular system (structural, chemical, and biological description), computational platform (hardware and software), and files (e.g. format, size, location).

The common data elements were then organized through a logical data model, which was refined to include dictionaries and minimize data redundancy. In this model the central concept is the virtual experiment, a set of dependent computational tasks represented by several input and output files. The organization of a virtual experiment as a list of processes and tasks was inspired by the CML-CompChem schema [23]. In CML-CompChem schema the job concept represents a computer simulation task that includes a series of consecutive sub-tasks designated as a job list. The concepts of experiment, process group, process, and task were introduced to handle the representation of tasks that might be run in parallel or sequentially, and that might target the same or different systems. The model also integrates various parameters related to MD, QM, and hybrid methods.

The data model defines a domain that should be sufficient to index biomolecular simulation data at the experiment level. In total over 60 classes were defined to represent the core data elements identified through the survey, along with new elements and dictionaries that should avoid data redundancy and facilitate queries using standard values. From a developer's prospective this model provides guidelines for the creation of a physical data model for the implementation of a database or an API. At a more abstract level the concepts introduced in this logical model provide a good starting point for the creation of a new terminology or ontology specific to biomolecular simulations.

One of the main features of this logical model is the integration of dictionaries to avoid data redundancy. For example a dictionary containing definitions of force fields (e.g. name, type, citations) can be referenced by molecular dynamics tasks, instead of creating individual force field definition entries every time the force field is used. The integration of dictionaries into the model does not enforce mappings to standard definitions but rather enable links between specific values and standard definitions if they exist.

The raw data used in the dictionaries was defined in various CSV files. Several scripts were created to either populate a database for remote queries or create an Apache Lucene [24] index for Java API searches that enable exact and partial string matching. The source code for the dictionaries, including the CSV files, SQL scripts, and Java API, is available from GitHub at: <https://github.com/jcvthibault/biosim-repository>.

Until this point the development of the dictionaries has been restricted to an internal effort by our group. Successful implementations of terminologies in other domains, such as the UMLS (Unified Medical Language System [25]), should be used to guide the organization of the raw data and facilitate the integration of existing terminologies representing particular aspects of the biomolecular simulations (e.g. chemical data, biomolecules, citations).

Ontologies

While the development of the data models and dictionaries described above represents a substantial advance for organizing biomedical simulations data, still this metadata infrastructure will not allow for complex queries of the data that may need the support of an ontology. An ontology is necessary to make possible complex query of the data that require additional reasoning using the underlying relationships existing among the different concepts in the data model and dictionaries. While in many cases the need of more complex queries may not be apparent to many researchers, a more detailed analysis of queries in multiple settings has demonstrated to us that complex queries supported by an ontology can be very useful. Our experience with building tools for complex distributed queries is that researchers tend to use cumbersome procedures requiring multiple independent queries followed by manual review of these results to obtain the information needed. In most cases our empirical observations have been that these cumbersome and time consuming methods can be replaced by appropriate ontology support. None-the-less we recognize that developing the “ultimate” ontology may not be feasible, therefore in this work we follow an incremental approach implementing only the relationships “is a” and “has part”. We will use a user center feedback approach to expand the relationships the BIOSIO will support in the future. To meet this need we have developed the BIOSIO that is reported here in detail for the first time.

The BIOSIO development follows the principles of the Open Biological and Biomedical Ontology (OBO) Foundry. BIOSIO builds upon the Basic Formal Ontology (BFO) [26] as its upper-level ontology. BFO defines abstract concepts such as “continuant”, i.e. an entity that exists and persists through time (e.g. a material entity, a spatial region), and “occurent”, i.e. an entity that has temporal parts (e.g. a process, an event, a temporal region). These concepts serve as a foundation for most OBO ontologies to facilitate interoperability and future developments. BIOSIO also builds upon more concrete ontologies derived from BFO: the Information Artifact Ontology (IAO), which describes information entities such as data sets, documents, software and algorithms, and the Ontology for Biomedical Investigations (OBI), which aims to describe the wide spectrum of biological and clinical investigations, from their design to the analysis methods and resulting data sets. [27] BIOSIO, like many other OBO ontologies, uses the ChEBI (Chemical Entity of Biological Interest) [28] ontology to define chemical and molecular entities, such as atoms, ions, molecules, nanostructures, nucleic acids, and proteins. Biological concepts can be derived from ChEBI by linking to other OBO ontologies, such as the Gene Ontology (GO) [29] or the Protein Ontology (PRO). [30] The BIOSIO thesaurus and ontology aim to represent the following concepts:

- Computational chemistry methods, including quantum chemistry and molecular dynamics.
- Analysis methods (e.g. Root mean square deviation calculations, principal component analysis).
- Computational tasks, including input and output description.
- Software packages and file formats.

The BIOSIO ontology was implemented in 3 different formats: as a relational database, as a Simple Knowledge Organization System (SKOS) and as an OWL 2 document). A database was first designed to store the concepts represented by the data model and dictionaries presented in previous section and validated by the user community as explained elsewhere [31]. The database schema (Figure 1) was inspired by the UMLS Metathesaurus [25]. Each concept (i.e. meaning) is defined in the CONCEPT table and can be associated to several terms (i.e. synonyms), citations, and textual descriptions. Relationships between concepts are defined in the RELATIONSHIP table, while types of relationships (e.g. “is a”, “has part”) are defined in the CONCEPT table and differentiated from regular concepts via the IS_REL flag. Concepts can also be mapped to concepts from external terminologies or ontologies via the CONCEPT_MAPPING table.

The SEMANTIC_TYPE table stores the various categories used to provide a high-level classification of all concepts in the thesaurus: the semantic types. Each concept can be associated to one or multiple semantic types via the CONCEPT_SEMANTIC_TYPE table. Just like in the UMLS, semantic types are defined to reduce the complexity of the thesaurus [32]. They can be used to group similar concepts together and facilitate searches and result filtering. For the design of this thesaurus we created a simple semantic network that would enable targeted searches based on the different parameters and methods (e.g., molecular dynamics vs. quantum chemistry) one could choose to setup the simulation.

A Java API was developed to enable the creation of SKOS and OWL documents from the thesaurus defined in the relational database. The API queries the database and iterate through all the concepts to write the associated triples into a SKOS or OWL Turtle file. The API can also be used to populate the database from a SKOS or OWL document, using the OWL API and the SKOS API. The following assumptions were made when developing the API: high-level relationships such as “is a”, and “has parts” are mapped using external ontologies that are assumed to be referenced in the thesaurus or the OWL ontology. In SKOS, hierarchical relationships are represented through the “narrower” and “broader” associations. For example “DNA” is a “broader” concept than “Nucleic acid”, and “all-atom molecular dynamics” is narrower than “molecular dynamics”.

The BIOSIO ontology builds upon the BFO and maps to the following OBO ontologies: IAO, OBI, and ChEBI. The final OWL document only references ontologies that can be explicitly imported via tools such as Protégé [33] if the associated concepts are necessary for the use case. The SKOS-encoded controlled vocabulary on the other hand does not include references to external sources.

One of the long-term goals for this thesaurus is to become part of a larger source of biomedical concepts, such as the UMLS, to provide new specialized concepts that did not exist previously. In order to evaluate the novelty of the concepts introduced in this thesaurus we compared the overlap between the UMLS concepts and the BIOSIO thesaurus concepts. A quantitative evaluation of this overlap was performed by looking at the matches between concept terms. An index of the UMLS thesaurus was created using Apache Lucene [24], which was used to check exact matches between normalized concept terms from the two sources. The Lucene index was built using the 2012AB UMLS release, which contains 9.8 million concept terms.

The normalized version of a term is obtained by removing common stop words and by using the canonical form of each word using the Lexical Variant Generator (LVG) [34] tool. This normalization step is performed on each UMLS term when building the index and on each thesaurus concept term that is looked up in the index. This process tends to reduce the number of false negatives when comparing strings. To facilitate the analysis of the matches proposed by our program each concept term in the index is associated to its CUI (Concept Unique Identifier), its original term, a normalized version of the term, and the source terminology for the concept (e.g. ICD-10, MESH, NCI).

In total 697 concepts and 870 associated terms are represented in the BIOSIO thesaurus and the core ontology. These concepts are organized through 677 “is a” relationships and 13 “has part” relationships, but the infrastructure develop here will allow the incorporation of more complex relationship as need arises. The thesaurus also includes 139 references (127 unique citations) and 12 semantic types organized in a network, as illustrated in Figure 2.

Out of the 697 BIOSIO thesaurus concepts, 94 had at least one term name that matched a UMLS metathesaurus concept name. Out of the 94 concepts, 33 concepts were mapped correctly, but most of these string matches were acronyms and did not actually represent the same concepts. For example the acronym SAS (surface-accessible surface) matched different gene names (NANS, TSPAN31) that use this string as alternate identifiers. Most of the true positives are related to software or hardware components (e.g. CPU, GPU, and file). This is expected since our thesaurus includes concepts related to the computing environment, but leave out the description of biomolecular systems, which would have great overlap with the UMLS. Another source of false positives is the difference in granularity between matching concepts. For example the concept “Analysis task” in our thesaurus really represents computational analysis tasks, and not a generic “analysis” (C0936012) or “analysis of substances” (C0002778).

The rationale behind the development of the proposed SKOS is to provide a controlled vocabulary for biomolecular simulations that can be used to describe published data using semantic web components. Our experience with iBIOMES [35] showed that complex queries cannot be built if the tags (i.e. metadata) associated to the published experiments do not have any semantic meaning. For example if a user is looking for all simulations that use molecular dynamics, one should expect the query engine to search for both classical and ab-initio MD simulations. This type of inference assumes the existence of a controlled vocabulary representing hierarchical relationships between available tags. iBIOMES (see

below) builds upon the iRODS [36] framework, which provides a distributed file system where files are indexed using Attribute-Value-Unit (AVU) triplets. One of the current directions undertaken by the iRODS developers is the integration of KOS within their indexing system by integrating HIVE (Helping Interdisciplinary Vocabulary Engineering) to manage and index SKOS-encoded controlled vocabularies. HIVE provides a core server to load SKOS documents and to enable keyword and SPARQL searches. HIVE also support automatic document tagging using key phrase extraction, based on the KEA (Keyphrase Extraction Algorithm [37]) tool. Assuming that a model is trained within KEA, this could enable automatic biomolecular simulation literature tagging and indexing.

Moving in this direction, we installed a local instance of HIVE and loaded our controlled vocabulary. The SKOS document was successfully validated using the online quality checker available at <http://qskos.poolparty.biz/> and loaded into HIVE. A screenshot the web interface of our local HIVE instance is presented in Figure 3.

In this section we presented a new thesaurus, BIOSIO, which focuses on the representation of the computational methods, parameters and environments that describe biomolecular simulations. A preliminary analysis was performed to check for overlaps between this thesaurus and the UMLS, one of the largest sources of biomedical concepts and shows only a modest overlap. Our results shows that a future integration of the BIOSIO thesaurus into the UMLS metathesaurus will require some manual work but semi-automatic mappings between concepts should be feasible. The current mapping algorithm, based on a simple index lookup, could be largely improved. For example one could remove acronyms from the automatic mapping step and rely only on expanded labels to compare strings. In our analysis we used the whole UMLS, although we are only interested in computational methods and computing environment components. To avoid false positives such as gene and protein names, we could filter out certain UMLS semantic types representing biomolecular and chemical entities, since these are not directly represented in our thesaurus and are not expected to match any concept.

A SKOS-encoded controlled vocabulary and a simple ontology were derived from the thesaurus. The SKOS was validated and loaded into HIVE to enable concept browsing and searches. The next steps will be to integrate HIVE into the iBIOMES repository architecture to enable data tagging using the controlled vocabulary and enable SPARQL searches for a future integration into the semantic web. The SKOS presented here is not sufficient to describe biomolecular simulations since the vocabulary does not include concepts related to molecular and chemical entities. Conversion tools such as skosify (<https://code.google.com/p/skosify/>) and the OBO-to-SKOS converter from the University of Manchester (<http://www.cs.man.ac.uk/~sjupp/skos/index.html>) could be used to generate a SKOS version of ChEBI and represent these missing pieces.

The derived ontology links to popular OBO ontologies to integrate detailed descriptions of biomolecule and chemical entities, but also for the integration of more abstract concepts that should facilitate its reuse in future OBO developments. Future directions include the integration of the BIOSIO ontology into the Chemical Information Ontology (CHEMINF) [38] which describes a domain that is similar to biomolecular simulations in many aspects.

The ontology presented here is very simple and could be improved by the inclusion of more associative relationships and axioms specific to the domain of biomolecular simulation to help inferring certain characteristics of computational experiments.

At this point the concepts and relationships defined in the thesaurus and the ontology have not been formally evaluated. This work was mostly done within the context of a single computational lab and did not involve outside experts. A survey could be used to receive general feedback but a detailed evaluation using divide-and-conquer approach might be more beneficial, but more complete methodologies for the evaluation of controlled vocabularies and ontologies could also be used [39, 40]. A detailed description of the ontology, the code and artifacts described here are available at <https://github.com/jcvthibault/biosio-vocab>.

Implementation of systems to store and share biomedical simulation data: iBIOMES and iBIOMES Lite

We have demonstrated the utility of the metadata infrastructure, data models, dictionaries and ontology described above by developing and testing two different systems that take advantage of the metadata infrastructure to archive in a searchable fashion data from biomolecular simulations. The two systems target two different scenarios: large-scale collaborations (iBIOMES) and research group data management or small-scale collaborations (iBIOMES Lite). Both systems use distribute computing technologies, because even for small collaborations the centralization of all the data can be a substantial challenge due to limitations in network bandwidth and/or local storage capabilities. Distributed systems offer a good solution for scientific research in general as data sources can be aggregated as a single resource despite being physically distant, and local control over the data at each node can be conserved. The Grid infrastructure [41] for example, provides storage, backup, and security solutions to data owners, and implements the necessary authentication and authorization mechanisms to expose data to the community or a specific audience. This is very important as researchers tend to be reluctant to expose all their data or give up ownership. Both infrastructures, iBIOMES and iBIOMES Lite, could also be implemented using a cloud infrastructure, but certain function in iBIOMES like its file transfer mechanisms may suffer serious performance degradation due the lack of support of efficient file transfer mechanisms like iRODS in most cloud infrastructures.

iBIOMES

The general architecture of iBIOMES is presented in Figure 4. At the lowest level, iRODS stores the file/collection metadata in a PostgreSQL database, and provides interfaces to manage the distributed resources integrated into the system. A MySQL database was added to store MD and QM related metadata definitions and dictionaries such as lists of force-fields, basis sets, software, etc. A Java API (ibiomes-core) was created to programmatically access iRODS resources and manage metadata that are specific to biomolecular simulations. This API, derived in part from our data model, also helps generating metadata by parsing the files that are being registered into the system in order to avoid manual annotation of the simulation data. Access to iRODS functionalities is facilitated through the Jargon Java API

provided by iRODS. Finally, a RESTful interface and a web portal provide access to the registered data in a more user-friendly fashion.

A REST interface was developed to offer web services for access to the metadata catalog and dictionaries. The metadata catalog is open access as it only contains general definitions of biomolecular simulation related metadata. The related services are mainly used to auto-complete user entries in the web interface (e.g. software name, custom metadata). The current web portal builds upon this REST interface and allows authenticated and authorized users to manage and search data registered in iBIOMES (Figure 5). Users can create queries based on the standard metadata catalog to retrieve simulations of interest. The queries can either target files or experiments (collections of files). A simple web interface is available to query data based on common attributes such as methods, molecule type (e.g. DNA, RNA, and protein) or residue chain (nucleotide or amino-acid sequence). Experiments can also be retrieved by simply entering keywords, in which case the metadata attribute is bypassed and the query only uses the value component of the metadata triplets to find matches. Advanced queries can be built as well. The user can pick and choose metadata attributes from the iBIOMES metadata catalog or manually enter user-specific attributes, then assign values to each attribute.

Matching experiments and files can be downloaded and data content can be summarized directly through different applets if the user has the right permissions. For example Jmol [42] is used for 3D rendering of molecules described in PDB, Mol2, MOL/SDF or GAUSSIAN log files (Figure 4c). 2D data such as time series in comma-separated or tab-delimited value format can be dynamically plotted through a Java applet based on the JFreeChart (<http://www.jfree.org/jfreechart/>) library (Figures 6a-b). Supported graphs include multi-line plots (e.g. comparison of RMSd of multiple runs) and heatmaps (2D-RMSd matrix). A “shopping cart” based on RENCi's iDrop applet (<https://code.renci.org/gf/project/irodsidrop>) also allows users to pick and choose files or collections of files they want to download in a bulk fashion (Figure 6d).

The publication process is facilitated by parsers that automatically generate metadata during file registration, and can be customized for the need of a particular user or lab through XML descriptors. Although our efforts have mainly focused on supporting AMBER and Gaussian datasets, we are currently working on improving our parsers for other popular MD and QM software packages, including GROMACS, CHARMM, Gaussian, GAMESS, and NWChem.

iBIOMES Lite

iBIOMES Lite's goal is to provide the means for researchers to index and summarize simulation data in restricted 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 folder or directory and subdirectories 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 (plots), 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, iBIOMES Lite does not provide access to the files associated to the published experiments. All files are categorized and listed, but only files presenting analysis data are made available for download.

The entry point for the web interface is a page listing all the published experiments (Figure 7) in the current instantiation of iBIOMES Lite. 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 to 4 different HTML pages. The summary page (Figure 7) presents a summary of the experiment protocol along with possible analysis data, plots and 3D structures, rendered via Jmol.[42] 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 8). A third HTML page provides a tree view that allows the user to browse the directory and subdirectories associated to the experiment and list their content (Figure 8).

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 folder 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 9). 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 [31]. The workflow and file tree structures are stored as XML files then transformed into several HTML pages via XSL (eXtensible Stylesheet Language). Plots are generated for analysis files when applicable, then stored in the iBIOMES Lite web directory along with the HTML files.

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.

The use of simple technology such as XSL transformation and HTML summaries makes it a viable solution in most environments. iBIOMES Lite was successfully deployed on several desktop computers and laptops, running different operating systems, but most importantly it was deployed at several US national computing centers where big data is generated every day.

Conclusions

In this paper we summarized our efforts toward the creation of collaborative platforms for researchers in the biomolecular simulation field, from the definition of requirements to the physical implementations of several infrastructures.

The model presented here is the first attempt to provide a common representation for biomolecular simulation experiments as a set of computational tasks that can use different levels of theories and different sets of parameters. The model is extensible and allows the representation of a wide variety of computational methods, including molecular dynamics, quantum chemistry and QM/MM. This model was successfully used for the design of file parsers that provide a software-independent representation of the computational experiments. Both the iBIOMES repository and iBIOMES Lite use these parsers to automatically generate common metadata and/or a logical representation of the experiments being published in these systems.

The data model was supplemented with a set of dictionaries to provide standard values and definitions for certain data elements such as computational method names. The data model and dictionaries were reorganized into a controlled vocabulary inspired by the UMLS metathesaurus. The controlled vocabulary introduces a new hierarchy between concepts and a set of semantic types to provide high-level categories for concept search and filtering. The controlled vocabulary was extended to a Simple Knowledge Organization System and a simple OWL ontology for use in a semantic web context. The ontology builds upon various OBO ontologies to enable interoperability with other popular biomedical ontologies.

iBIOMES is the first open architecture for a distributed repository enabling biomolecular simulation data sharing. The system can be deployed by researchers at their own sites, independently from the computational resources that are used to generate the data. Data are exposed to the repository through a simple publication process. Data exchange is enabled by indexing the raw data via common data elements while collaboration is enabled through authentication and authorization mechanisms to protect private data and open public datasets to anonymous users.

While the iBIOMES repository provides a distributed solution to biomolecular simulation data storage and indexing, other solutions are needed to manage data hosted in limited settings where the user does not have root privileges or access to IT support to deploy database components. iBIOMES Lite was specifically designed to be deployed and used in these limited settings. This is the first effort aiming at summarizing data at the source, whether it is on a personal laptop or at national computing centers with thousands of computational nodes.

Acknowledgements

JCF was partially funded by NCATS Award Number 1ULTR001067. JCT and TEC acknowledge NSF support from CHE-1266307. Computational support was provided by the Center for High Performance Computing (CHPC) at the University of Utah.

References

1. Dror RO, Dirks RM, Grossman JP, Xu H, Shaw DE. Biomolecular simulation: A computational microscope for molecular biology. *Annu. Rev. Biophys.* 2002; 41:429–452. [PubMed: 22577825]
2. Alonso H, Bliznyuk AA, Gready JE. Combining docking and molecular dynamic simulations in drug design. *Med. Res. Rev.* 2006; 26 Authors: Provide first and last pages.

3. Klein ML, Shinoda W. Large-scale molecular dynamics simulations of self-assembling systems. *Science* (Washington D C). 2008; 321:5890.
4. Case DA, Cheatham TE 3rd, Darden T, Gohlke H, Luo R, Merz KM Jr. Onufriev A, Simmerling C, Wang B, Woods RJ. The Amber biomolecular simulation programs. *J. Comput. Chem.* 2005; 26:1668–1688. [PubMed: 16200636]
5. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* 2005; 26:1781–802. [PubMed: 16222654]
6. Brooks BR, Brucoleri RE, Olafson BD, Swaminathan S, Karplus M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* 2004; 4:187–217.
7. Hess B. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *Abstracts of Papers American Chemical Society.* 2009; 237
8. Plimpton S. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* 1995; 117:1–19.
9. Frisch, MJ.; Trucks, GW.; Schlegel, HB.; Scuseria, GE.; Robb, MA.; Cheeseman, JR.; Montgomery, JA., Jr; Vreven, T.; Kudin, KN.; Burant, JC. Gaussian 03, revision D. 01. Gaussian, Inc; Wallingford, CT: 2004.
10. Valiev M, Bylaska EJ, Govind N, Kowalski K, Straatsma TP, Van Dam HJJ, Wang D, Nieplocha J, Apra E, Windus TL. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Commun.* 2010; 181:1477–1489.
11. Schmidt MW, Baldrige KK, Boatz JA, Elbert ST, Gordon MS, Jensen JH, Koseki S, Matsunaga N, Nguyen KA, Su S. General atomic and molecular electronic structure system. *J. Comput. Chem.* 2004; 14:1347–1363.
12. Kong J, White CA, Krylov AI, Sherrill D, Adamson RD, Furlani TR, Lee MS, Lee AM, Gwaltney SR, Adams TR. Q-Chem 2.0: a high-performance ab initio electronic structure program package. *J. Comput. Chem.* 2000; 21:1532–1548.
13. Schrödinger, LLC. Jaguar, version 7.5. New York, NY: 2008.
14. Ng MH, Johnston S, Wu B, Murdock SE, Tai K, Fangohr H, Cox SJ, Essex JW, Sansom MSP, Jeffreys P. BioSimGrid: Grid-Enabled Biomolecular Simulation Data Storage and Analysis. *Future Gener. Comp. Syst.* 2006; 22:657–664.
15. Simms AM, Toofanny RD, Kehl C, Benson NC, Daggett V. Dynameomics: Design of a computational lab workflow and scientific data repository for protein simulations. *Protein Eng. Des. Sel.* 2008; 21:369–377. [PubMed: 18411223]
16. Alfredsson M. presented at the Proc. UK eScience All Hands Meeting. 2005 unpublished.
17. Calleja M, Bruin R, Tucker MG, Dove MT, Tyer R, Blanshard L, Van Dam KK, Allan RJ, Chapman C, Emmerich W. Collaborative grid infrastructure for molecular simulations: The eMinerals minigrid as a prototype integrated compute and data grid. *Molec. Simul.* 2005; 31:303–313.
18. Meyer T, D'Abramo M, Hospital A, Rueda M, Ferrer-Costa C, Perez A, Carrillo O, Camps J, Fenollosa C, Repchevsky D, Lluís Gelpi J, Orozco M. MoDEL (molecular dynamics extended library): A database of atomistic molecular dynamics trajectories. *Structure.* 2010; 18:1399–1409. [PubMed: 21070939]
19. Hospital A, Andrio P, Fenollosa C, Cicin-Sain D, Orozco M, Lluís Gelpi J. MDWeb and MDMoby: An integrated web-based platform for molecular dynamics simulations. *Bioinformatics.* 2012; 28:1278–1279. [PubMed: 22437851]
20. Beveridge DL, Cheatham TE III, Mezei M. The ABCs of molecular dynamics simulations on B-DNA, circa 2012. *J. Biosci. (Bangalore).* 2012; 37:379–97.
21. Lavery R, Zakrzewska K, Beveridge D, Bishop TC, Case DA, Cheatham T III, Dixit S, Jayaram B, Lankas F, Laughton C, Maddocks JH, Michon A, Osman R, Orozco M, Perez A, Singh T, Spackova N, Sponer J. A systematic molecular dynamics study of nearest-neighbor effects on base pair and base pair step conformations and fluctuations in B-DNA. *Nucleic acids Res.* 2010; 38:299–313. [PubMed: 19850719]

22. Thibault JC, Roe DR, Facelli JC, Cheatham TE. Data model, dictionaries, and desiderata for biomolecular simulation data indexing and sharing. *J. Cheminf.* 2014;4. Authors: Provide missing information.
23. [Oct 3, 2006] Chemical Markup Language or CML: an XML based language for describing molecules. <http://www.xml-cml.org/>
24. Apache Lucene Foundation. 2014. from <https://lucene.apache.org/>
25. Bodenreider O. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Res.* 2004; 32:D267–D270. [PubMed: 14681409]
26. Grenon P, Smith B, Goldberg L. Biodynamic ontology: applying BFO in the biomedical domain. *Stud. Health Technol. Inf.* 2004; 102:20–38.
27. Brinkman RR, Courtot M, Derom D, Fostel JM, He Y, Lord P, Malone J, Parkinson H, Peters B, Rocca-Serra P, Ruttenberg A, Sansone SA, Soldatova LN, Stoeckert CJ Jr. Turner JA, Zheng J, consortium OBI. Modeling biomedical experimental processes with OBI. *J. Biomed. Sem.* 2010; 1(Suppl 1):S7.
28. Hastings J, de Matos P, Dekker A, Ennis M, Harsha B, Kale N, Muthukrishnan V, Owen G, Turner S, Williams M, Steinbeck C. The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Res.* 2013; 41:D456–D463. [PubMed: 23180789]
29. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nature Gen.* 2000; 25:25–29.
30. Natale DA, Arighi CN, Barker WC, Blake JA, Bult CJ, Caudy M, Drabkin HJ, D'Eustachio P, Evisikov AV, Huang H. The Protein Ontology: a structured representation of protein forms and complexes. *Nucleic Acids Res.* 2011; 39:D539–545. [PubMed: 20935045]
31. Thibault JC, Cheatham TE, Facelli JC. IBIOMES lite: Summarizing biomolecular simulation data in limited settings. *J. Chem. Inf. Model.* 2014; 54:1810–1819. [PubMed: 24830957]
32. McCray AT, Burgun A, Bodenreider O. Aggregating UMLS semantic types for reducing conceptual complexity. *Stud. Health Technol. Inform.* 2001:216–220. Authors: Provide volume number. [PubMed: 11604736]
33. Gennari JH, Musen MA, Ferguson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Tu SW. The evolution of Protégé: An environment for knowledge-based systems development. *Intern. J. Hum.-Comput. Stud.* 2003; 58:89–123.
34. SKOS. [July 2015] <http://www.w3.org/2004/02/skos/>
35. Thibault JC, Facelli JC, Cheatham TE. IBIOMES: Managing and sharing biomolecular simulation data in a distributed environment. *J. Chem. Inform. Model.* 2013; 53:726–736.
36. Rajasekar A, Moore R, Hou C, Lee CA, Marciano R, de Torcy A, Wan M, Schroeder W, Chen SY, Gilbert L. iRODS Primer: Integrated rule-oriented data system. *Synth. Lect. Inform. Conc., Retrieval, Serv.* 2010; 2:1–143.
37. Medelyan, O.; Witten, IH. presented at the Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries; 2006. unpublished
38. Hastings J, Chepelev L, Willighagen E, Adams N, Steinbeck C, Dumontier M. The chemical information ontology: provenance and disambiguation for chemical data on the biological semantic web. *PloS one.* 2011; 6:e25513. [PubMed: 21991315]
39. Cimino JJ. Desiderata for controlled medical vocabularies in the twenty-first century. *Meth. Inf. Med.* 1998; 37:394–403.
40. Bright TJ, Yoko Furuya E, Kuperman GJ, Cimino JJ, Bakken S. Development and evaluation of an ontology for guiding appropriate antibiotic prescribing. *J. Biomed. Inf.* 2012; 45:120–128.
41. Foster, I.; Kesselman, C. The grid 2: Blueprint for a new computing infrastructure. Morgan Kaufmann; 2003.
42. Herráez A. Biomolecules in the computer: Jmol to the rescue. *Biochem. Mol. Biol. Educ.* 2006; 34:255–261. [PubMed: 21638687]

Figure 1.
Database schema for the BIOSIO thesaurus.

Figure 2.
BIOSIO thesaurus semantic network.

Figure 3.
Screenshot of the Hive web interface for SKOS concept browsing used in BIOSIO.

Figure 4.

General architecture of iBIOMES. At the lowest level, iRODS stores the file metadata while a separate MySQL database enforces standard metadata use. A REST interface and a web client provide query and update capability to the metadata catalog through the iRODS API (Jargon) and an iBIOMES-specific API (ibiomes-core). (From [35]).

Figure 5. iBIOMES web interface. (a) Summary page for a Molecular Dynamics simulation of RNA. (b) Search results with key features for each experiment (software, number of atoms, type of molecule, author, etc).

Figure 6. Applets used in the iBIOMES web interface for data visualization and bulk downloads. A plot applet based on the JFreeChart library enables comparison of multiple RMSd (root mean square deviation) plots (a) and rendering of RMSd 2D matrices as heatmaps (b). Jmol is used to render and manipulate 3D structures (c). The iDrop Lite applet is used for bulk downloads of files through the shopping cart service (d). (From [35]). Figure 6 has to be printed in color

Method	Name	Software	Molecules	Publisher	Date	Experiment path
1BIVM1	AMBER	<i>R11A / Protein</i>	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/amber/1BIVM1
ZH49	AMBER	<i>R11A</i>	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/amber/ZH49
A-DNA	AMBER	<i>D11A</i>	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/amber/tutorial1/a-dna
ACAC	GAUSSIAN	$C_8H_7O_2$	Juji	2014-01-08 14:34	/home/juji/ibiomes/test/gaussian/acac
ACETONITRILE	GAMESS	C_2H_3N	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/gamess/acetonitrile
CH5N_NBO	NWChem	CH_3N	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/nwchem/ch5n_nbo
CLX2-MULTI	GAUSSIAN	$C_{60}H_{12}O_6$	Juji	2014-01-08 14:34	/home/juji/ibiomes/test/gaussian/clx2-multi
DFT_BSSE	NWChem	H_2O	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/nwchem/dft_bsse
EMBNET-TUTORIAL	CHARMM	<i>Protein</i>	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/charmm/embnet-tutorial
ETHANOL	NWChem	<i>Biomolecule</i>	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/nwchem/ethanol
FAD	GROMACS	$C_{27}H_{39}O_{10}P_2$	Juji	2014-01-08 14:35	/home/juji/ibiomes/test/gromacs/FAD
FOSFINA	GAUSSIAN	$C_{18}H_{18}P$	Juji	2014-01-08 14:34	/home/juji/ibiomes/test/gaussian/fosfina
GLI	GAUSSIAN	$C_2H_4IO_2$	Juji	2014-01-08 14:34	/home/juji/ibiomes/test/gaussian/gli

Figure 7.
Listing of published experiments in iBIOMES Lite website. (From [31]).

Figure 8.
Summary of an experiment within the iBIOMES Lite website.

Figure 9.
iBIOMES Lite publication process. (From [31]).