BMC Bioinformatics

SOFTWARE Open Access

Native structure-based modeling and simulation of biomolecular systems per mouse click

Benjamin Lutz[1,2], Claude Sinner[1,2], Stefan Bozic[1], Ivan Kondov[1*] and Alexander Schug[1]

Abstract

Background: Molecular dynamics (MD) simulations provide valuable insight into biomolecular systems at the atomic level. Notwithstanding the ever-increasing power of high performance computers current MD simulations face several challenges: the fastest atomic movements require time steps of a few femtoseconds which are small compared to biomolecular relevant timescales of milliseconds or even seconds for large conformational motions. At the same time, scalability to a large number of cores is limited mostly due to long-range interactions. An appealing alternative to atomic-level simulations is coarse-graining the resolution of the system or reducing the complexity of the Hamiltonian to improve sampling while decreasing computational costs. Native structure-based models, also called Gō-type models, are based on energy landscape theory and the principle of minimal frustration. They have been tremendously successful in explaining fundamental questions of, e.g., protein folding, RNA folding or protein function. At the same time, they are computationally sufficiently inexpensive to run complex simulations on smaller computing systems or even commodity hardware. Still, their setup and evaluation is quite complex even though sophisticated software packages support their realization.

Results: Here, we establish an efficient infrastructure for native structure-based models to support the community and enable high-throughput simulations on remote computing resources via GridBeans and UNICORE middleware. This infrastructure organizes the setup of such simulations resulting in increased comparability of simulation results. At the same time, complete workflows for advanced simulation protocols can be established and managed on remote resources by a graphical interface which increases reusability of protocols and additionally lowers the entry barrier into such simulations for, e.g., experimental scientists who want to compare their results against simulations. We demonstrate the power of this approach by illustrating it for protein folding simulations for a range of proteins.

Conclusions: We present software enhancing the entire workflow for native structure-based simulations including exception-handling and evaluations. Extending the capability and improving the accessibility of existing simulation packages the software goes beyond the state of the art in the domain of biomolecular simulations. Thus we expect that it will stimulate more individuals from the community to employ more confidently modeling in their research.

Keywords: Protein folding, RNA folding, Native structure-based model, Molecular dynamics, GridBeans

Background

Great progress in experimental techniques, such as X-ray diffraction analysis and nuclear magnetic resonance spectroscopy, has led to a vastly increased diversity and quality of biomolecular structure data presented in the Protein Data Bank [1]. By combining this information with biomolecular simulation one can supplement static structural models with an increasingly detailed dynamic picture even for huge molecular machines like the ribosome [2,3]. Still, exploring the dynamical nature of molecular life poses a significant challenge for present-day computational resources. While astonishing progress in this field has led to first all-atom protein folding simulations for small globular proteins on the millisecond timescale [4], the required specialized supercomputers are not publicly available.

*Correspondence: ivan.kondov@kit.edu
[1] Steinbuch Centre for Computing, Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
Full list of author information is available at the end of the article

An intriguing alternative to simulating these biomolecules with atomic resolution in physics-/chemistry-based forcefields is focusing on their essential features and coarse-graining (CG) either the resolution of the system or its forcefield [5,6]. In such CG models, the granularity of the system is typically changed from an all-atom representation by mapping groups of atoms into single beads. This reduces the computational complexity and provides access to longer timescales and length scales. For example, in the MARTINI forcefield typically four heavy atoms and the associated hydrogens are mapped into a single bead representing the respective group [7]. Another approach, the so-called native structure-based modeling (SBM) [8-11], is based on energy landscape theory and the principle of minimal frustration [12]. Accordingly, the energy landscape has a funnel-like shape biased towards the native state of a protein. In a long evolutionary process, the energy landscape was smoothened by minimizing (energetic) roughness and frustration to enable efficient folding by ensuring a dominance of native interactions over non-native ones. Thus the structural information of the native state becomes an integral part of the model potential describing the interactions in the biomolecular system. Topological information, such as the contact map of the native state, is usually employed (Gō potentials [13]) initially within coarse-grained C_α [14] or C_β [15] models and more recently, within all-atom models [16]. The introduced bias towards the native state reduces the forcefield complexity without loss of essential information and enables the simulation on biologically relevant timescales, e.g. protein folding simulation [16] on the all-atom level on standard desktop computers. In many recent studies of protein dynamics SBM has become the tool of choice to rationalize experimental observations by means of computer simulations [17]. Structure based modeling provides now an established method for physical understanding of, e.g. folding pathways [14,18], folding kinetics [19], effects like posttranslational modifications [20] or oligomerization [21]. The SBM has been generalized to describe also proteins with two or more stable native conformations [22] in order to model functional transitions, e.g. allostery and ligand binding. Moreover, transition states, which are experimentally directly not accessible, have been studied using SBM [23,24]. Still, one might easily expect that focusing on native interactions and neglecting non-native interactions within SBM distorts simulation results. Significant effort has therefore been invested to examine to which extend non-native interactions play critical roles [17,25-28]. In particular, recent work has carefully analyzed the role of native interactions in prior atomically resolved simulations [29,30]. These studies have found dominance of native interactions to non-native ones and good agreement between results from CG SBM simulations and more fine-grained models [31]. Overall, SBM is accurate for the extreme case of minimal or no frustration, with all-atom models realizing all possible non-native interactions [32]. Somewhat surprisingly, neglecting non-native interactions does not significantly distort simulations results [17,31]. This not only makes the minimalistic SBM very useful, but also suggests that naturally occurring proteins seem to possess low frustration [8,33].

In numerous close collaborations between experimental and modeling research groups the SBM methods are applied [34-36]. To enable regular use by the community of, in particular, experimental scientists or other researchers who do not possess specialized programming and/or modeling experience, it seems senseful to establish a research infrastructure (similar to the PDB service) standardizing and simplifying the simulation setup and submission, as well as the evaluation of these simulations. This infrastructure should include services for development of novel models and adaption of existing models to new applications, and routine deployment of ready-to-go models. A first effort to establish such a service is the SMOG (Structure-based MOdels in GROMACS) web server [37] that is publicly available under http://smog-server.org/. This server provides a convenient setup of native structure-based simulations with several options for custom forcefield choice and parameterization. Going beyond mere folding, eSMBTools [38] is a Python-based toolkit allowing the simplified setup and evaluation of native structure-based simulations for proteins and RNA. The focus of this toolkit is enhancing these simulations by experimental or bioinformatics derived data, e.g. to enable the prediction of protein complexes or active conformations based on the statistical analysis of existing sequence databases [34,39] or riboswitch folding [40]. In particular, forcefields and topologies of amino and nucleic acids are encoded in XML files making the toolkit easily extensible to other biomolecules, such as ligands.

Providing a modeling and simulation service for SBM solves several challenging issues which we outline in the following: i) The simulations require use of computing resources which are usually unavailable locally and the scientist has to face the high technical complexity of distributed computing infrastructure. To this end, several solutions providing access to remote computing resources already exist [41-43]. However, while hiding the complexity via virtualization of resources and abstraction, these middleware services are rather generic so that their direct use without the integration of the biomolecular model can pose even higher barriers for the end-user. ii) Multiple program codes (steps) have to be linked together in one composite application (workflow) via standard interfaces for automatic execution. Data sources and sinks at different workflow steps have to be linked via standard interfaces (dataflow). The existing solutions do not use generic standards (such as for example web services) but rather domain-specific solutions which have to be laboriously adapted to every new model and simulation. Currently, there are many program codes that do not blend in with each other and therefore efforts have been recently spent to partially alleviate this problem [44-46]. iii) The elements of the infrastructure exposed to researchers have to be reduced to minimum and made available via a modern graphical user interface (GUI). The challenging aspect is here the design decision what has to be included in the interface rather than the GUI implementation itself. The access to more functions improves the tool capabilities and flexibility but heightens the expert level.

In previous work, many of these issues have been tackled effectively for applications in high-throughput virtual screening [44], materials science [45,47] and biomolecular NMR [46]. In particular, a data model based framework for data exchange between workflow steps has been proposed and a toolkit has been provided for automatic generation of a data access service for scientific applications [48]. Thereby, the issues outlined above can be treated by means of modern technologies such as web services and model-driven engineering leading to complete automation of the program interfaces, workflows and dataflows. A common approach which has been extensively applied is the science gateway (also known as web portal). For example, the virtual research community WeNMR has a large collection of portals providing production services for different applications in structural biology [49], including molecular dynamics simulations with Gromacs [50]. Data exchange in multi-step molecular simulations and analysis has been treated in several works previously [51-54]. Within the MoSGrid project the molecular simulation markup language (MSML) [51,52] has been developed employing the concept of Chemical Markup Language (CML) dictionaries and used in quantum chemistry, molecular dynamics and docking simulations on the MoSGrid portal [55]. The Collaborative Computing Project for NMR (CCPN) [53] has provided a software framework that consists of a data model [54], the CcpNmr Analysis program, and a collection of additional tools, including the CcpNmr FormatConverter. The CCPN application programming interface (API) is available in three programming languages (C, Python and Java) and enables the integration of additional analysis and simulation software to build complex workflow applications.

In this paper, we present a software infrastructure which deploys SBM on distributed high-throughput computing (HTC) and high performance computing (HPC) resources providing a powerful interface for model development and user-friendly interface. The software provides a simple and still flexible graphical user interface for eSBMTools to allow end users to run SBM simulations without developing IT technical skills.

Implementation

We have adopted the principles of Service Oriented Architecture (SOA) [56] to design the implementation of the platform. Thus, many of the generic components required for the implementation are available in existing and well established grid and cloud middleware stacks, from which we have selected the UNICORE middleware [41]. UNICORE is a fully fledged and mature open-source middleware which has been deployed and supported on large computing infrastructures such as PRACE (http://www.prace-ri.eu/) in Europe and more recently also on XSEDE (https://www.xsede.org/) in the USA. Computing clusters and other HPC resources managed by common batch systems, such as SGE, LSF, PBS Torque and LoadLeveler, can readily be used with UNICORE. Currently, UNICORE offers four different client variants: command line client, graphical rich client, a portal client and a high-level API. The UNICORE Rich Client (URC) [57] provides us with the software basis to integrate eSBMTools within an application extension called GridBean [58]. In addition, UNICORE includes a workflow engine and a powerful graphical workflow editor, completely integrated in the service layer and the URC, respectively.

Furthermore, there are several alternative open source middleware solutions. In the following, we briefly review two of them. The Globus Toolkit [42] is a widely used open source toolkit which implements numerous standards (for example web services) and allows building infrastructures for grid (internet) computing [59]. It has overall standards conformity similar to that of UNICORE but does not provide graphical clients and portals and can be used with the Galaxy workflow system [60]. The middleware ARC (Advanced Resource Connector) [43] has been developed and included in the software stack of the European Middleware Initiative. It implements web services standards for server-client communication and provides a GUI client. ARC integrates with the third-party workflow engine Taverna [61].

Implementation architecture

Figure 1 shows the overall architecture of our eSBMTools integration. The main component is the SBM GridBean, a Java-based component, that captures and validates the SBM simulation parameters provided by the user via a GUI. The URC is the runtime container for the GridBean and provides the basic functionality to access UNICORE services e.g. for submitting and monitoring

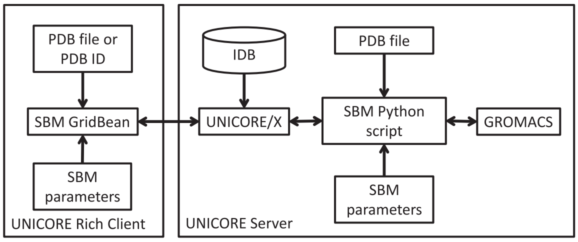


Figure 1 General architecture of SBM simulation with eSBMTools based on UNICORE.

jobs, managing file storage and handling authentication and authorization.

The implementation of the SBM GridBean is based on the GridBean API [58] and consists of three major parts:

- A configuration file `gridbean.xml` which defines runtime parameters for the GridBean in the URC, e.g. names and versions of the GridBean and of the target application on the server;
- A Java class containing the GridBean model which defines the job parameters and input/output files;
- A plugin class which defines the graphical user interface and the mapping of the input components to the GridBean model. Here, a validator can be defined which performs type and plausibility checks on input parameters.

During job submission the URC translates the input from the SBM GridBean (model parameters, files, variables, required computing resources) into a JSDL (Job Submission Description Language) request which is then sent together with a PDB input file or a PDB ID to the UNICORE server. The UNICORE server has an incarnation database (IDB) which determines how to handle the incoming JSDL request. The IDB includes entries for all applications that are available to the URC for job submission. The IDB entry for the SBM application defines the Python interpreter as job executable and several parameters to configure the SBM Python script which is introduced in the next subsection.

SBM Python script

The SBM Python script is based on the Python toolkit eSBMTools [38] that provides a wide range of functionalities to setup and manipulate structure-based models and to evaluate simulation output. Along with modular functions, Python is platform-independent and readily available on most HPC systems. Therefore it is an excellent choice for the functionality that eSBMTools is aiming for. The script consists of various preprocessing and post-processing modules that include functions called by a central Python script. This Python script represents the functional core unit of the SBM GridBean. The toolkit interfaces with GROMACS, a molecular dynamics software package [62] in a version provided by the SMOG homepage [63] that features an extension called `g_kuh`. The GROMACS extension `g_kuh` calculates the number of formed native contacts within the structure for each dumped frame of the simulated trajectory. The number of formed native contacts is often referred to as the Q value.

Figure 2 shows a block diagram that depicts the utilized functionalities of the SBM Python script. The user provides a PDB ID [1] of the protein of interest and simulation options. The PDB ID is passed to the module `PdbFile` that prompts an according coordinate file download from the database. The PDB coordinate file and the provided options are processed by module `GoModel` that generates the required files for an SBM forcefield and atom coordinates in supported GROMACS file formats. Simulation options, e.g. temperature, simulation steps, random generation seeds, etc., are passed to module `MdpFile` that generates the simulation configuration file accordingly. After the GROMACS simulation the protein's trajectory is evaluated by modules `TopFile` and `Qvalues` that create plots of the protein's contact map and Q values trajectory, respectively. The contact map is a two-dimensional representation of the residue-residue contacts present in the native conformation and the Q value trajectory is the temporal evolution of the number of formed native contacts along the simulated trajectory.

Results and discussion

The SBM GridBean

Based on the SBM Python script (see previous section) we have developed an SBM GridBean that allows users to configure and run SBM simulations. Using UNICORE technology, we do not have to handle user

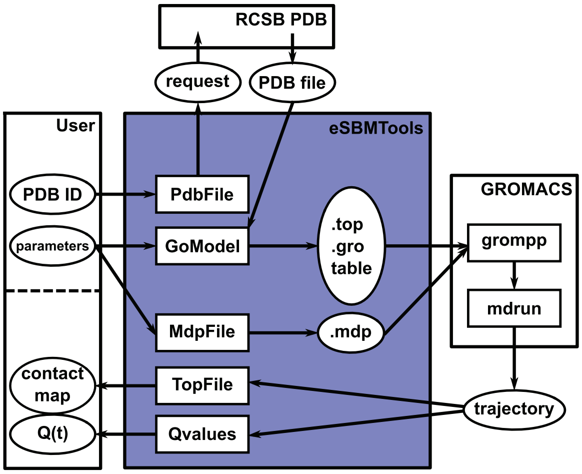


Figure 2 Functionalities of the SBM Python script.

authentication and authorization, web service protocols for job submission or file protocols during development but can rather concentrate on building an intuitive GridBean GUI which provides input fields for several methods and parameters for eSBMTools. The GridBean also validates all values entered by a user before sending them to the server. The GUI contains several tabs which group the parameters together (see Figure 3). In the following we outline these tabs.

PDB parameters

The PDB tab (see Figure 3, left screenshot) specifies the molecular structure to be analyzed. The user has the choice to specify a PDB ID in a text field. The PDB file with the actual molecular structure data is then downloaded by the SBM Python script. Alternatively, a PDB file which is available on the local file system can be specified. During job submission the PDB file is copied to the UNICORE server. In both cases the user can select the protein chains which should be processed by eSBMTools. By default all chains are selected.

Molecular dynamics parameters (MDP)

The MDP tab, shown in Figure 3 in the middle, offers more general parameters to control the `mdrun` program. The tab is structured into the three categories "Run control", "Initialization" and "Output control". These parameters are mapped by the Python script to an `*.mdp` file which forms the input for the GROMACS preprocessor `grompp`.

Forcefield parameters

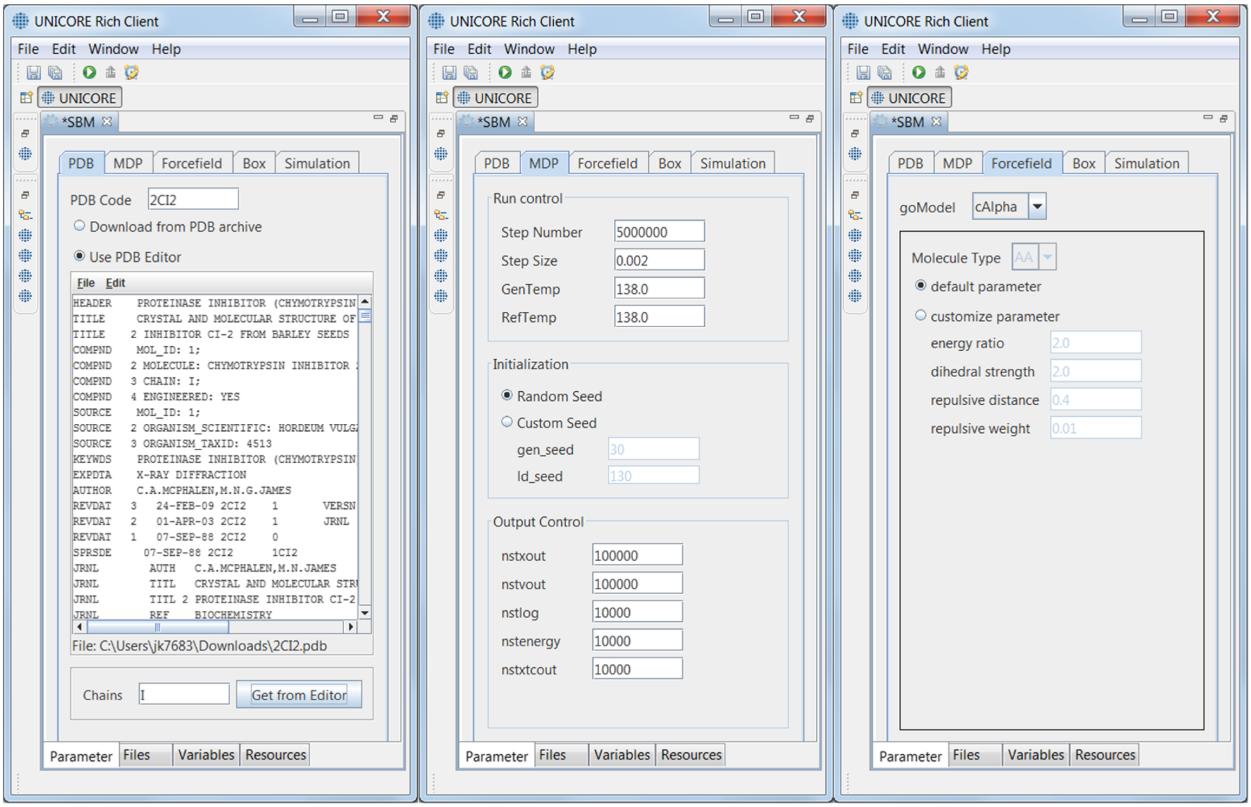
The "Forcefield" tab defines specific parameter regarding the forcefield of an SBM simulation. The user can choose between a coarse grained C_α or an all-atom model which has influence on the precision and the runtime of the job. At the all-atom level the user has the ability to choose between amino acid (AA) and nucleic acid (NA) as molecule type. Depending on the chosen molecule type the corresponding topologies of molecular building blocks for proteins (AA) or DNA/RNA (NA) are used. A screenshot of this tab can be seen in Figure 3, on the right.

Box parameters

The distance between the outer most atoms of the molecule and the rectangular simulation box in all three dimensions can be adjusted on this panel.

Simulation parameters

The simulation tab allows to specify whether the structure-based model is only prepared and returned or a


UNICORE Rich Client
File Edit Window Help
UNICORE
*SBM
PDB MDP Forcefield Box Simulation
Parameter Files Variables Resources

PDB Code 2CI2

○ Download from PDB archive

● Use PDB Editor

File Edit

```
HEADER    PROTEINASE INHIBITOR (CHYMOTRYPSIN
TITLE     CRYSTAL AND MOLECULAR STRUCTURE OF
TITLE    2 INHIBITOR CI-2 FROM BARLEY SEEDS
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: CHYMOTRYPSIN INHIBITOR 
COMPND   3 CHAIN: I;
COMPND   4 ENGINEERED: YES
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: HORDEUM VULG
SOURCE   3 ORGANISM_TAXID: 4513
KEYWDS    PROTEINASE INHIBITOR (CHYMOTRYPSIN
EXPDTA    X-RAY DIFFRACTION
AUTHOR    C.A.MCPHALEN,M.N.G.JAMES
REVDAT   3   24-FEB-09 2CI2    1       VERSN
REVDAT   2   01-APR-03 2CI2    1       JRNL
REVDAT   1   07-SEP-88 2CI2    0
SPRSDE     07-SEP-88 2CI2      1CI2
JRNL        AUTH   C.A.MCPHALEN,M.N.JAMES
JRNL        TITL   CRYSTAL AND MOLECULAR STRU
JRNL        TITL 2 PROTEINASE INHIBITOR CI-2
JRNL        REF    BIOCHEMISTRY
```

File: C:\Users\jk7683\Downloads\2CI2.pdb

Chains I Get from Editor

Run control

Step Number 5000000
Step Size 0.002
GenTemp 138.0
RefTemp 138.0

Initialization

● Random Seed
○ Custom Seed
gen_seed 30
ld_seed 130

Output Control

nstxout 100000
nstvout 100000
nstlog 10000
nstenergy 10000
nstxtcout 10000

goModel cAlpha

Molecule Type AA
● default parameter
○ customize parameter
energy ratio 2.0
dihedral strength 2.0
repulsive distance 0.4
repulsive weight 0.01

Figure 3 Screenshots displaying the input parameter tabs "PDB", "MDP" and "Forcefield" (from left to right, respectively) of the graphical user interface of the SBM GridBean. Only the GridBean view is displayed, all other views of the URC (including the Grid Browser) are minimized.

simulation with the structure-based model, i.e. the actual simulation is started on the computing resource (such as an HPC cluster) attached to the UNICORE server. In the first case the SBM Python script will create only the input files `*.mdp`, `*.gro` and `*.top` files and in case of a C_α simulation the file `table.xvg`. Creating only the input files is useful for computing sites where GROMACS is not available or where the system resources are limited to perform a computationally demanding `mdrun`. The created simulation files can then be transferred to a more capable computing site with a GROMACS installation. In the second case the SBM Python script creates all configuration files and calls `grompp` and `mdrun`. Both GROMACS commands are started as separate processes. The results of this simulation type are plots of the contact map and Q values as function of time.

Further functions implemented and used in URC

Monitoring

An important feature of the URC is the Grid Browser with which the status of submitted simulations can be monitored. For every submitted simulation (single job or workflow) a working directory is created. This directory is the execution environment of the SBM Python script and contains the generated simulation files and the output of `grompp` and `mdrun`. The files in the working directory can be viewed and downloaded within the Grid Browser.

Update site

Another benefit of using UNICORE is the straightforward installation of GridBeans into the URC. As the URC is based on Eclipse it comes with an integrated update mechanism. For instance, the SBM GridBean is installed by specifying the URL[a] of the project's update site and following the instructions of the setup wizard.

Integration of third-party libraries

By using the Java based GridBean API and Eclipse as base technologies for the SBM GridBean we have integrated further Java libraries from the domain of bioinformatics into the SBM GridBean, for example the Jmol [64] library for visualization tasks. The structure of the simulated PDB or the trajectories from GROMACS are visualized with Jmol. BioJava [65] is another library that we have integrated into the SBM GridBean. Local PDB files (from the PDB tab) can be parsed and the value of the chains parameter is then automatically filled in using BioJava.

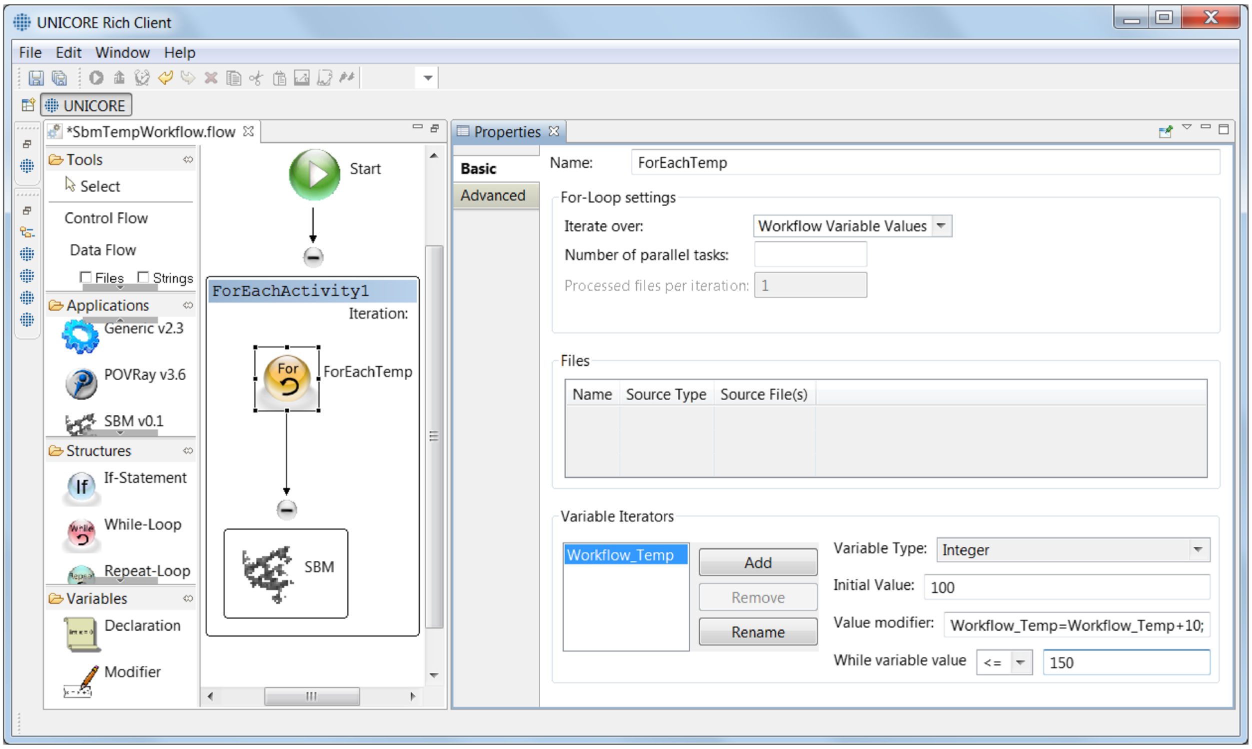
Workflows

GridBeans are reusable components which can be integrated into composite models in the form of UNICORE workflows [57,66]. With the graphical workflow editor, which is standard component in the URC, a graph can be built specifying the execution order (control flow) of the simulation steps using several different GridBeans from the "Applications" pane of the URC. An output file of a GridBean can be transferred as input file to another GridBean (dataflow). The job submission and the file handling is done automatically by the UNICORE workflow service. In the next section we introduce a detailed case study of a workflow that includes the SBM GridBean.

Exemplary workflow (case study)

To provide a concrete application example that employs the developed SBM GridBean, we present a case study of protein folding dynamics of the before described application scenarios within a basic UNICORE workflow. We process ten exemplary proteins (PDB IDs 2CI2, 1G6P, 1ENH, 1SHF, 2QJL, 1RYK, 1RIS, 1BTH, 1TEN, 1MJC) containing from 45 up to 99 amino acids and both pure alpha-helical and mixed alpha-helical/beta-sheet structures, giving a cross-sectional overview. The SBM GridBean facilitates the setup and execution of an MD simulation in GROMACS and two exemplary evaluation steps at a specific temperature: A contact map is generated and the Q values along the simulated trajectory are calculated. To this end, the SBM GridBean is embedded in a *foreach* workflow control structure (see Figure 4) which automatically executes this step for each temperature. An analysis like this can be used, e.g., in algorithms that search for folding temperatures of proteins. In this study each protein was simulated at six different temperatures (100, 110, 120, 130, 140, and 150 in reduced GROMACS units, see the "Properties" tab in Figure 4) which enclose the region of expected folding temperatures in the present SBM parametrization. The folding temperature characterizes the temperature at which folded and unfolded conformations are equally occupied during a simulation. The constructed workflow is submitted and the simulation progress is monitored in the Grid Browser shown in Figure 5 on the left. The Jmol molecule viewer (see Figure 5 on the right) and further Eclipse plugins, that are integrated in the URC, allow visualization of the simulation results. For each temperature the workflow generates the contact map of the protein and a plot of the Q value trajectory as a function of time, depicted in Figures 6 and 7, respectively. The contact map gives detailed structural information about the protein's native state. Based on the Q value trajectory it is possible to estimate whether the protein is in its folded or unfolded state at the simulated temperature.

The case study demonstrates the practicability of the presented SBM GridBean in operation on 10 exemplary proteins. The GridBean provides reusability for arbitrary protein structures at desired temperatures which allows its direct integration into workflows. The end



Name: ForEachTemp

For-Loop settings
Iterate over: Workflow Variable Values
Number of parallel tasks:
Processed files per iteration: 1

Variable Iterators
Workflow_Temp
Variable Type: Integer
Initial Value: 100
Value modifier: Workflow_Temp=Workflow_Temp+10;
While variable value <= 150

Figure 4 Workflow for finding the protein's folding temperature which is constructed in the workflow editor of URC employing the SBM GridBean. The setup of the *foreach* control structure used for running several SBM simulations for different temperatures is done in the "Properties" tab, displayed on the right. Only the workflow editor view of the URC is shown.

user is not confronted with the details of the model or the implementation itself but can focus on the design and execution of the desired studies. The technical challenges are transferred to a developer who has carried out the required core implementation (SBM GridBean, SBM Python script). This core implementation needs to meet the requirements of projected workflows for which it might be beneficial in the future to split up the GridBean in parts dealing with pre- and post-processing.

In the Additional files we provide a screen dump showing the installation and setup processes (Additional file 1), as well as the usage of the SBM GridBean for the case study discussed above (Additional files 2 and 3). In this case study, we make use of the pilot service that is currently available for employees and students at KIT. In future, we plan to provide such a service for broader community as part of e-infrastructure projects.

Benefits and drawbacks

In the following we will provide an outline of the major benefits of using our proposed software tools combined with a critical discussion of the drawbacks, particularly in comparison with existing alternative solutions.

The GUI of the SBM GridBean provides intuitive access to the most common methods of the eSBMTools modules and enables a wide range of individuals to run SBM simulations. While an end-user of the SBM GridBean are not faced with any line of code, the flexibility of changing the internal logic of the simulation steps is limited compared to the usage of the eSBMTools API directly. Thus, some variations of the model would require relevant changes in the SBM Python script. Nevertheless, this restricted flexibility has an additional advantage because the GUI does not expose well documented and validated features for changes the end-user. This increases the overall quality and reproducibility of the simulation output.

Generating the input files for GROMACS via a web server is a useful approach. The software must be installed only once on the web server and is then accessible from all over the world. By using the SBM GridBean the user has to additionally install the URC on their local desktop and the SBM GridBean into the URC from an update-site. If the web server is not attached to a computing cluster it may have limited resources for MD runs. In these cases, the prepared input files can be transferred from the web server to a more capable computing infrastructure that provides generic services for MD simulations with GROMACS. However, the UNICORE service comes with an integrated solution to access modern HPC and HTC computing infrastructures and is not

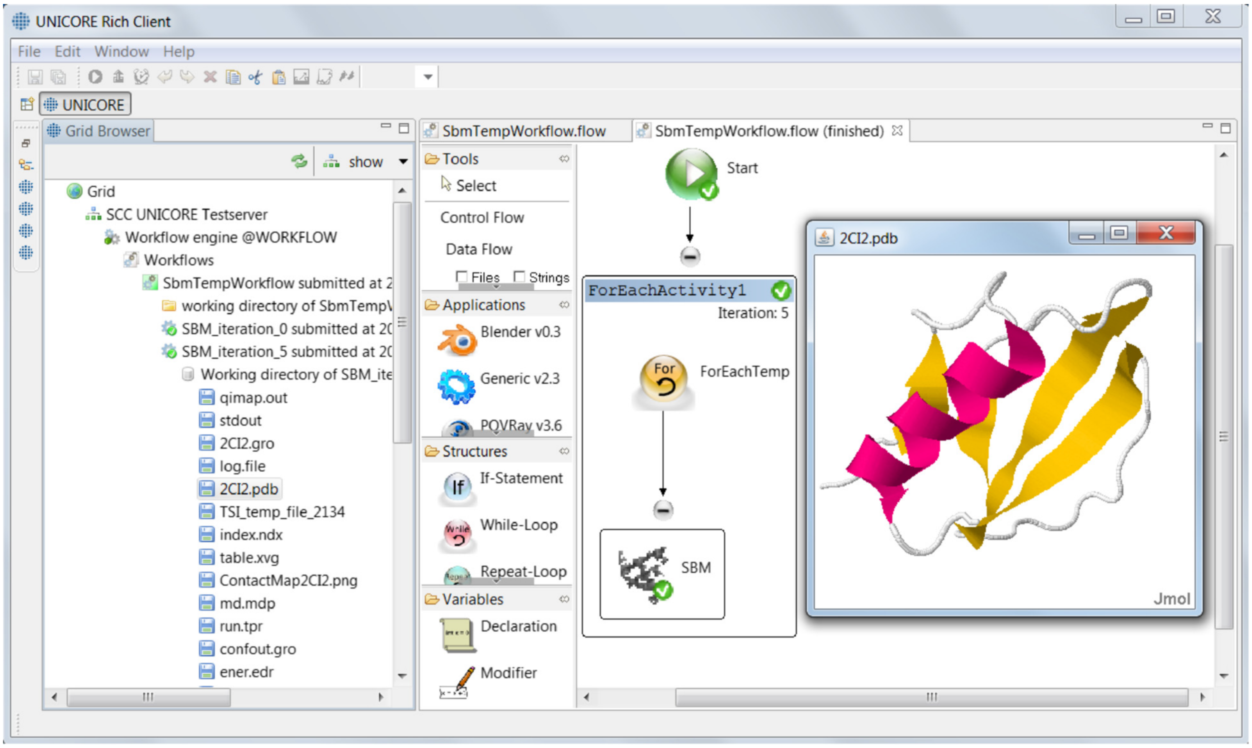


Figure 5 The Grid Browser and further plugins, components of the URC, allow showing the progress and the results of the simulation. On the left pane the GridBrowser is displayed, in the middle the finished workflow and the Jmol in the small window on the right.

only capable to prepare the input files for the simulation but also to efficiently execute computationally demanding all-atom SBM simulations using a massively parallel version of GROMACS. In all cases, the end-user will benefit from the uniform environment for modeling and simulation setup provided by the URC and the SBM GridBean.

Users who use a web server are supposed to trust the service providers in respect of handling their data. In addition to encrypting the whole client-server communication via SSL, the middleware UNICORE uses X.509 certificates for authentication and thus can ensure that only authorized persons have access to the connected resources. While contributing to the overall security substantially, managing X.509 certificates is considered generally more complex compared to simple user credentials such as username and password which are currently not supported. We expect that in future UNICORE will provide alternative authentication mechanisms.

In the following, we compare our proposed new software to an established tool in the community, particularly to the SMOG server, which was already introduced above. Except for the source code extensions to GROMACS it is a closed-source system leading to different concepts for establishing trust relations with their end-users compared to an open-source product. Furthermore, the extension of the platform with further functionalities, e.g. connecting to computing resources, and the setup an own instance of the service is not possible. In contrast, the eSBMTools API and the SBM GridBean are open source. Interested parties (end-users but mostly service providers) can download,

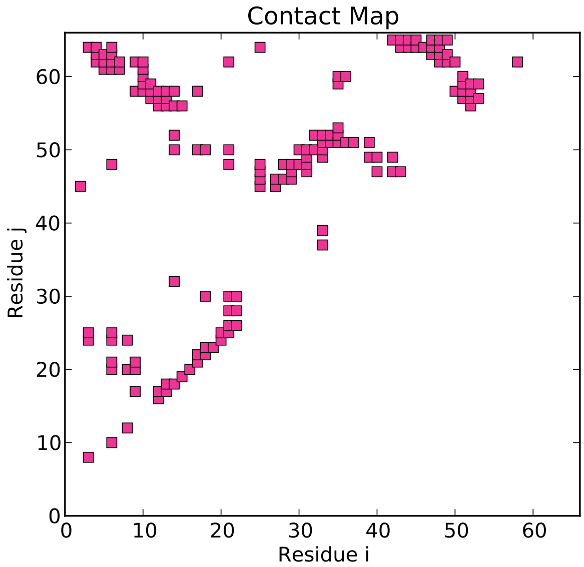


Figure 6 Contact map of the serine proteinase inhibitor CI-2 with PDB ID 2CI2.

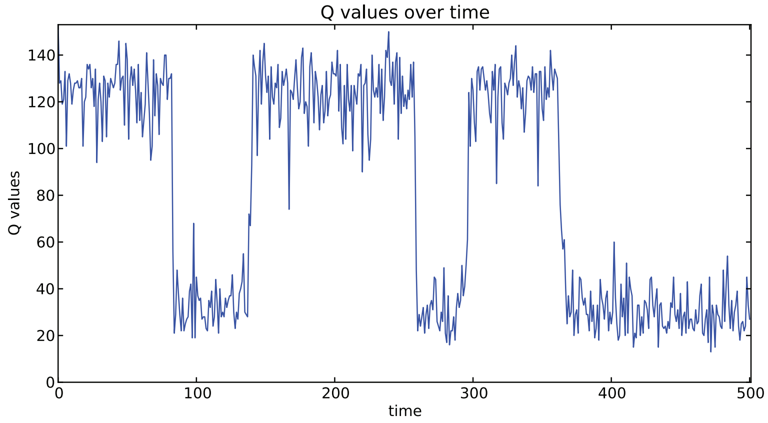


Figure 7 Q value trajectory of a simulation for the protein CI-2. The transitions between Q ≈ 40 and Q ≈ 120 indicate folding and unfolding events.

adapt, redistribute and productively use the source code for their purposes.

The eSBMTools API and the SBM GridBean make use of several well known and tested bioinformatics libraries such as numpy, biopython, Jmol etc. These third-party libraries are well tested and have a high quality by permanent observation and development within the community. Using them increases the quality of the software and enriches it with many useful features for the end-user. Although Java and Python are used as the programming languages for the implementation of eSBMTools and the SBM GridBean, no programming language knowledge is required for using the SBM GridBean in the URC for constructing workflow models and running simulations.

The functionalities for constructing and executing workflows using UNICORE enables the design of individual custom-made projects employing SBM of biomolecular systems. The laborious working steps and protocols, as well as security mechanisms are hidden in the inner logic of the URC, the SBM GridBean and the UNICORE service and only properties and functions relevant for modeling and execution of workflows are exposed through the user interface so that end-users can focus on solving domain-specific challenges in biophysics, biochemistry or bioinformatics.

In Table 1 we summarize the benefits and potential drawbacks of our implementation of SBM compared to the SMOG server, eSBMTools and the SBM GridBean, as discussed above.

Conclusions

Significant progress on the technological side and the development of increasingly accurate forcefields enable biomolecular simulations which provide atomically detailed insight into the molecular machinery of life, yet require expert knowledge for the setup and analysis of data. One common class of such biomolecular simulations, native structure-based or Gō-type models, contributes to answer questions ranging from protein and RNA folding to function and structure prediction. We have developed a framework to facilitate construction and execution of workflows for these simulations based on the UNICORE middleware. We showed the straightforward setup of an exemplary workflow and expect that it can be adapted to individual projects as a service for the biomolecular simulation community.

Availability and requirements

- **Project name:** UNICORE based integration of eSBMTools
- **Project home page:** The home page of eSBMTools is http://sourceforge.net/projects/esbmtools. The source code of the SBM GridBean is available under http://www.multiscale-modelling.eu/svn/esbmtools/gridbeans.

Table 1 Comparison between SMOG server and SBM GridBean

Feature	SMOG server	eSBMTools	SBM GridBean
Graphical user interface	Yes	No	Yes
Access via web browser	Yes	No	No
Access via web server	Yes	No	Yes
Access to distributed computing resources	No	No	Yes
Use of X.509 certificate	No	No	Yes
Open source	No	Yes	Yes
Integration of third-party libraries	?	Yes	Yes
Provide means for custom modeling	No	Yes	Yes
Integration in composite models	No	Yes	Yes

- **Operating system(s):** Platform independent
- **Programming language:** Java and Python
- **Other requirements:** UNICORE (version 6) server is required on the server host, URC (version 6) on the client host, Java Runtime Environment on both the client and server hosts, and Python interpreter and GROMACS on the computing resource.
- **License:** FreeBSD license (2-clause BSD license) for the SBM GridBean (Java source code) and GNU GPL (General Public License) for eSBMTools (Python source code)

Endnote

[a]For this project, the public update site is http://www.multiscale-modelling.eu/update-site/esbmtools/0.1/.

Additional files

Additional file 1: Demonstration of installation and usage of the SBM GridBean: Part 1. This video shows how to add an UNICORE site to the Grid Browser and install the SBM GridBean from an Update Site.

Additional file 2: Demonstration of installation and usage of the SBM GridBean: Part 2. This video shows how to construct a workflow for finding the folding temperature of a protein.

Additional file 3: Demonstration of installation and usage of the SBM GridBean: Part 3. This video shows how to submit and monitor the simulation and view the results.

Abbreviations

API: Application programming interface; CML: Chemical markup language; GUI: Graphical user interface; HPC: High performance computing; HTC: High throughput computing; IDB: Incarnation data base; IT: Information technology; JSDL: Job submission description language; MD: Molecular dynamics; MDP: Molecular dynamics parameter; MSML: Molecular simulation markup language; NMR: Nuclear magnetic resonance; PDB: Protein data bank; RNA: Ribonucleic acid; SBM: Structure based modeling; SMOG: Structure-based MOdels in GROMACS; SOA: Service oriented architecture; SSL: Secure socket layer; URC: UNICORE Rich Client; XML: Extensible markup language.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

BL and CS provided their knowledge on eSBMTools, participated in the design of the GridBean, carried out the case studies (workflow simulations), and drafted the manuscript. SB has programmed the SBM GridBean and drafted the manuscript. IK coordinated the project, participated in the conception and worked on finalizing the manuscript; AS conceived of the study, and participated in its design and coordination and helped to finalize the manuscript. All authors read and approved the final manuscript.

Acknowledgements

A.S. acknowledges support from the *Impuls- und Vernetzungfonds* of the Helmholtz Association. This work has been partially funded by the 7th Framework Programme of the European Commission within the Research Infrastructures with grant agreement number RI-261594, project MMM@HPC.

Author details

[1]Steinbuch Centre for Computing, Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany. [2]Department of Physics, Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany.

Received: 2 June 2014 Accepted: 22 August 2014
Published: 29 August 2014

References

1. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The protein data bank.** *Nucleic Acids Res* 2000, **28:**235–242. [http://dx.doi.org/10.1093/nar/28.1.235]
2. Whitford PC, Geggier P, Altman RB, Blanchard SC, Onuchic JN, Sanbonmatsu KY: **Accommodation of aminoacyl-tRNA into the ribosome involves reversible excursions along multiple pathways.** *RNA* 2010, **16**(6):1196–1204.
3. Bock LV, Blau C, Schröder GF, Davydov II, Fischer N, Stark H, Rodnina MV, Vaiana AC, Grubmüller H: **Energy barriers and driving forces in tRNA translocation through the ribosome.** *Nat Struct Mol Biol* 2013, **20:**1390–1396. [http://dx.doi.org/10.1038/nsmb.2690]
4. Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, Bank JA, Jumper JM, Salmon JK, Shan Y, Wriggers W: **Atomic-level characterization of the structural dynamics of proteins.** *Science* 2010, **330**(6002):341–346.
5. Voth GA (Ed): *Coarse-Graining of Condensed Phase and Biomolecular Systems*: CRC Press; 2008.
6. Schug A, Hyeon C, Onuchic JN: **Coarse-grained structure-based simulations of proteins and RNA.** In *Coarse-Graining of Condensed Phase and Biomolecular Systems*. Voth GA. Boca Raton: CRC Press Taylor & Francis, Group; 2008:123–140. Chapter 9.
7. de Jong DH, Singh G, Bennett WFD, Arnarez C, Wassenaar TA, Schäfer LV, Periole X, Tieleman DP, Marrink SJ: **Improved parameters for the martini coarse-grained protein force field.** *J Chem Theor Comput* 2013, **9:**687–697. [http://dx.doi.org/10.1021/ct300646g]
8. Onuchic JN, Wolynes PG: **Theory of protein folding.** *Curr Opin Struct Biol* 2004, **14:**70–75. [http://www.sciencedirect.com/science/article/pii/S0959440X04000107]
9. Noel JK, Onuchic JN: **The many faces of structure-based potentials: from protein folding landscapes to structural characterization of complex biomolecules.** In *Computational Modeling of Biological Systems,* Biological and Medical Physics, Biomedical Engineering. Edited by Dokholyan NV. US: Springer; 2012:31–54. [http://dx.doi.org/10.1007/978-1-4614-2146-7_2]
10. Whitford PC, Sanbonmatsu KY, Onuchic JN: **Biomolecular dynamics order-disorder transitions and energy landscapes.** *Rep Progr Phys* 2012, **75**(7):076601. [http://stacks.iop.org/0034-4885/75/i=7/a=076601]
11. Schug A, Onuchic JN: **From protein folding to protein function and biomolecular binding by energy landscape theory.** *Curr Opin Pharmacol* 2010, **10**(6):709–714.
12. Bryngelson JD, Onuchic JN, Socci ND, Wolynes PG: **Funnels, pathways, and the energy landscape of protein folding: a synthesis.** *Protein Struct Funct Bioinformatics* 1995, **21**(3):167–195. [http://dx.doi.org/10.1002/prot.340210302]
13. Go N: **Protein folding as a stochastic process.** *J Stat Phys* 1983, **30**(2):413–423. [http://dx.doi.org/10.1007/BF01012315]
14. Clementi C, Nymeyer H, Onuchic JN: **Topological and energetic factors what determines the structural details of the transition state ensemble and "en-route" intermediates for protein folding? An investigation for small globular proteins.** *J Mol Biol* 2000, **298**(5):937–953. [http://www.sciencedirect.com/science/article/pii/S0022283600936933]
15. Oliveira LC, Schug A, Onuchic JN: **Geometrical features of the protein folding mechanism are a robust property of the energy landscape: a detailed investigation of several reduced models.** *J Phys Chem B* 2008, **112**(19):6131–6136.
16. Whitford PC, Noel JK, Gosavi S, Schug A, Sanbonmatsu KY, Onuchic JN: **An all-atom structure-based potential for proteins: bridging minimal models with all-atom empirical forcefields.** *Protein Struct Funct Bioinformatics* 2009, **75**(2):430–441. [http://dx.doi.org/10.1002/prot.22253]
17. Sinner C, Lutz B, John S, Reinartz I, Verma A, Schug A: **Simulating biomolecular folding and function by native-structure-based/go-type models.** *Isr J Chem* 2014. [http://dx.doi.org/10.1002/ijch.201400012]
18. Rey-Stolle MF, Enciso M, Rey A: **Topology-based models and NMR structures in protein folding simulations.** *J Comput Chem* 2009, **30**(8):1212–1219. [http://dx.doi.org/10.1002/jcc.21149]
19. Chavez LL, Onuchic JN, Clementi C: **Quantifying the roughness on the free energy landscape: entropic bottlenecks and protein folding**

rates. *J Am Chem Soc* 2004, **126**(27):8426–8432. [http://dx.doi.org/10.1021/ja049510+]

20. Shental-Bechor D, Levy Y: **Effect of glycosylation on protein folding: a close look at thermodynamic stabilization.** *Proc Natl Acad Sci* 2008, **105**(24):8256–8261. [http://www.pnas.org/content/105/24/8256.abstract].
21. Klein P, Mattoon D, Lemmon MA, Schlessinger J: **A structure-based model for ligand binding and dimerization of EGF receptors.** *Proc Natl Acad Sci U S A* 2004, **101**(4):929–934. [http://www.pnas.org/content/101/4/929.abstract]
22. Lammert H, Schug A, Onuchic JN: **Robustness and generalization of structure-based models for protein folding and function.** *Protein Struct Funct Bioinformatics* 2009, **77**(4):881–891. [http://dx.doi.org/10.1002/prot.22511]
23. Clementi C, Jennings PA, Onuchic JN: **Prediction of folding mechanism for circular-permuted proteins.** *J Mol Biol* 2001, **311**(4):879–890.
24. Schug A, Whitford PC, Levy Y, Onuchic JN: **Mutations as trapdoors to two competing native conformations of the Rop-dimer.** *Proc Natl Acad Sci* 2007, **104**(45):17674–17679.
25. Li L, Mirny LA, Shakhnovich EI: **Kinetics, thermodynamics and evolution of non-native interactions in a protein folding nucleus.** *Nat Struct Biol* 2000, **7**(4):336–342. [http://dx.doi.org/10.1038/74111]
26. Karanicolas J, Brooks CL III: **Improved go-like models demonstrate the robustness of protein folding mechanisms towards non-native interactions.** *J Mol Biol* 2003, **334**(2):309–325. [http://www.sciencedirect.com/science/article/pii/S0022283603011999]
27. Clementi C, Plotkin SS: **The effects of non-native interactions on protein folding rates: theory and simulation.** *Protein Sci* 2004, **13**(7):1750–1766. [http://dx.doi.org/10.1110]
28. Clementi C: **Coarse-grained models of protein folding: toy models or predictive tools?.** *Curr Opin Struct Biol* 2008, **18:**10–15. [http://www.sciencedirect.com/science/article/pii/S0959440X07001753] [Folding and Binding/Protein-nucleic acid interactions].
29. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE: **How fast-folding proteins fold.** *Science* 2011, **334**(6055):517–520. [http://www.sciencemag.org/content/334/6055/517.abstract]
30. Shaw DE, Maragakis P, Lindorff-Larsen K, Piana S, Dror RO, Eastwood MP, Bank JA, Jumper JM, Salmon JK, Shan Y, Wriggers W: **Atomic-level characterization of the structural dynamics of proteins.** *Science* 2010, **330**(6002):341–346.
31. Best RB, Hummer G, Eaton WA: **Native contacts determine protein folding mechanisms in atomistic simulations.** *Proc Natl Acad Sci* 2013, **110**(44):17874–17879. [http://www.pnas.org/content/110/44/17874.abstract]
32. Plotkin SS: **Speeding protein folding beyond the go model: how a little frustration sometimes helps.** *Protein Struct Funct Bioinformatics* 2001, **45**(4):337–345. [http://dx.doi.org/10.1002/prot.1154]
33. Ferreiro DU, Hegler JA, Komives EA, Wolynes PG: **Localizing frustration in native proteins and protein assemblies.** *Proc Natl Acad Sci* 1981, **104**(50):9–19824. [http://www.pnas.org/content/104/50/19819.abstract]
34. Dago AE, Schug A, Procaccini A, Hoch JA, Weigt M, Szurmant H: **Structural basis of histidine kinase autophosphorylation deduced by integrating genomics, molecular dynamics, and mutagenesis.** *Proc Natl Acad Sci* 2012, **109**(26):E1733–E1742.
35. Nechushtai R, Lammert H, Michaeli D, Eisenberg-Domovich Y, Zuris JA, Luca MA, Capraro DT, Fish A, Shimshon O, Roy M, Schug A, Whitford PC, Livnah O, Onuchic JN, Jennings PA: **Allostery in the ferredoxin protein motif does not involve a conformational switch.** *Proc Natl Acad Sci* 2011, **108**(6):2240–2245.
36. Mickler M, Dima RI, Dietz H, Hyeon C, Thirumalai D, Rief M: **Revealing the bifurcation in the unfolding pathways of GFP by using single-molecule experiments and simulations.** *Proc Natl Acad Sci* 2026, **104**(51):8–20273.
37. Noel JK, Whitford PC, Sanbonmatsu KY, Onuchic JN: **SMOG@ctbp: simplified deployment of structure-based models in GROMACS.** *Nucleic Acids Res* 2010, **38**(suppl 2):W657–W661. [http://nar.oxfordjournals.org/content/38/suppl_2/W657.abstract]
38. Lutz B, Sinner C, Heuermann G, Verma A, Schug A: **eSBMTools 1.0 enhanced native structure-based modeling tools.** *Bioinformatics* 2013, **29**(21):2795–2796. [http://dx.doi.org/10.1093/bioinformatics/btt478]
39. Schug A, Weigt M, Onuchic JN, Hwa T, Szurmant H: **High-resolution protein complexes from integrating genomic information with molecular simulation.** *Proc Natl Acad Sci* 2009, **106**(52):22124–22129.
40. Lutz B, Faber M, Verma A, Klumpp S, Schug A: **Differences between cotranscriptional and free riboswitch folding.** *Nucleic Acids Res* 2013, **42**(4):2687–2696.
41. Streit A, Bala P, Beck-Ratzka A, Benedyczak K, Bergmann S, Breu R, Daivandy J, Demuth B, Eifer A, Giesler A, Hagemeier B, Holl S, Huber V, Lamla N, Mallmann D, Memon A, Memon M, Rambadt M, Riedel M, Romberg M, Schuller B, Schlauch T, Schreiber A, Soddemann T, Ziegler W: **UNICORE 6 — recent and future advancements.** *Ann Telecommunications* 2010, **65:**757–762. [http://dx.doi.org/10.1007/s12243-010-0195-x]
42. Foster I, Kesselman C: **Globus: a metacomputing infrastructure toolkit.** *Int J High Perform Comput Appl* 1997, **11**(2):115–128. [http://hpc.sagepub.com/content/11/2/115.abstract]
43. Ellert M, Grønager M, Konstantinov A, Kónya B, Lindemann J, Livenson I, Nielsen J, Niinimäki M, Smirnova O, Wäänänen A: **Advanced resource connector middleware for lightweight computational grids.** *Future Generat Comput Syst* 2007, **23**(2):219–240. [http://www.sciencedirect.com/science/article/pii/S0167739X06001178]
44. Pérez-Sánchez H, Kondov I, García JM, Klenin K, Wenzel W: **A pipeline pilot based SOAP implementation of FlexScreen for high-throughput virtual screening.** In *Proceedings of the 3rd International Workshop on Science Gateways for Life Sciences (IWSG-Life 2011), London, United Kingdom, June 8–10, 2011, Volume 819 of CEUR-WS.org*. Edited by Terstyanszky G, Kiss T. CEUR; 2011:9. [http://ceur-ws.org/Vol-819/]
45. Kondov I, Maul R, Bozic S, Meded V, Wenzel W: **UNICORE-based integrated application services for multiscale materials modelling.** In *UNICORE Summit 2011 Proceedings, 7–8 July 2011, Torun, Poland, Volume 9 of IAS Series*. Edited by Romberg M, Bala P, Müller-Pfefferkorn R, Mallmann D: Jülich: Forschungszentrum Jülich GmbH Zentralbibliothek; 2011:1–10. [http://hdl.handle.net/2128/4518]
46. Schneider O, Fogh RH, Sternberg U, Klenin K, Kondov I: **Structure simulation with calculated NMR parameters — integrating COSMOS into the CCPN framework.** In *HealthGrid Applications and Technologies Meet Science Gateways for Life Sciences, Volume 175 of Studies in Health Technology and Informatics*. Edited by Gesing S, Glatard T, Krüger J, Olabarriaga SD, Solomonides T, Silverstein JC, Montagnat J, Gaignard A, Krefting D: IOS Press; 2012:162–172. [http://dx.doi.org/10.3233/978-1-61499-054-3-162]
47. Bozic S, Kondov I, Meded V, Wenzel W: **UNICORE-based workflows for the simulation of organic light-emitting diodes.** In *UNICORE Summit 2012 Proceedings, May 30–31, 2012, Dresden, Germany, Volume 15 of IAS Series*. Edited by Huber V, Müller-Pfefferkorn R, Romberg MR: Jülich: Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag; 2012:15–25. [http://hdl.handle.net/2128/4705]
48. Bender A, Poschlad A, Bozic S, Kondov I: **A service-oriented framework for integration of domain-specific data models in scientific workflows.** *Procedia Comput Sci* 2013, **18:**1087–1096. [http://dx.doi.org/10.1016/j.procs.2013.05.274], [2013 International Conference on Computational Science]
49. Wassenaar TA, van Dijk M, Loureiro-Ferreira N, Schot G, Vries SJ, Schmitz C, Zwan J, Boelens R, Giachetti A, Ferella L, Rosato A, Bertini I, Herrmann T, Jonker HRA, Bagaria A, Jaravine V, Güntert P, Schwalbe H, Vranken WF, Doreleijers JF, Vriend G, Vuister GW, Franke D, Kikhney A, Svergun DI, Fogh RH, Ionides J, Laue ED, Spronk C, Jurkša S, et al: **WeNMR: Structural Biology on the Grid.** *J Grid Comput* 2012, **10**(4):743–767. [http://dx.doi.org/10.1007/s10723-012-9246-z]
50. van Dijk M, Wassenaar TA, Bonvin AM: **A flexible, grid-enabled web portal for GROMACS molecular dynamics simulations.** *J Chem Theor Comput* 2012, **8**(10):3463–3472. [http://pubs.acs.org/doi/abs/10.1021/ct300102d]
51. Birkenheuer G, Blunk D, Breuers S, Brinkmann A, dos Santos Vieira I, Fels G, Gesing S, Grunzke R, Herres-Pawlis S, Kohlbacher O, Kruger J, Lang U, Packschies L, Muller-Pfefferkorn R, Schafer P, Steinke T, Warzecha KD, Wewior M: **MoSGrid: efficient data management and a standardized data exchange format for molecular simulations in a grid environment.** *J Cheminformatics* 2012, **4**(Suppl 1):P21. [http://dx.doi.org/10.1186/1758-2946-4-S1-P21]

52. Grunzke R, Breuers S, Gesing S, Herres-Pawlis S, Kruse M, Blunk D, de la Garza L, Packschies L, Schäfer P, Schärfe C, Schlemmer T, Steinke T, Schuller B, Müller-Pfefferkorn R, Jäkel R, Nagel WE, Atkinson M, Krüger J: **Standards-based metadata management for molecular simulations.** *Concurrency Comput Pract Ex* 2013. [http://dx.doi.org/10.1002/cpe.3116]
53. Fogh RH, Boucher W, Vranken WF, Pajon A, Stevens TJ, Bhat TN, Westbrook J, Ionides JMC, Laue ED: **A framework for scientific data modeling and automated software development.** *Bioinformatics* 2005, **21**(8):1678–1684. [http://bioinformatics.oxfordjournals.org/content/21/8/1678.abstract]
54. Vranken WF, Boucher W, Stevens TJ, Fogh RH, Pajon A, Llinas M, Ulrich EL, Markley JL, Ionides J, Laue ED: **The CCPN data model for NMR spectroscopy: development of a software pipeline.** *Protein Struct Funct Bioinformatics* 2005, **59**(4):687–696. [http://dx.doi.org/10.1002/prot.20449]
55. Grunzke R, Birkenheuer G, Blunk D, Breuers S, Brinkmann A, Gesing S, Herres-Pawlis S, Kohlbacher O, Krüger J, Kruse M, Müller-Pfefferkorn R, Schäfer P, Schuller B, Steinke T, Zink A: **A data driven science gateway for computational workflows.** In *UNICORE Summit 2012, Dresden, Germany, Volume 15 of IAS Series*: Jülich: Forschungszentrum, Jülich GmbH Zentralbibliothek; 2012:35–49.
56. **Service-oriented architecture (SOA).** Published online [http://en.wikipedia.org/wiki/Service-oriented_architecture] Accessed 2012-12-07.
57. Demuth B, Schuller B, Holl S, Daivandy J, Giesler A, Huber V, Sild S: **The UNICORE Rich Client: facilitating the automated execution of scientific workflows.** In *2010 IEEE Sixth International Conference on e-Science (e-Science), Brisbane, QLD*: IEEE; 2010:238–245.
58. Ratering R, Lukichev A, Riedel M, Mallmann D, Vanni A, Cacciari C, Lanzarini S, Benedyczak K, Borcz M, Kluszczynski R, Bala P, Ohme G: **GridBeans: supporting e-science and grid applications.** In *Second IEEE International Conference on e-Science and Grid Computing, 2006 (e-Science '06), Amsterdam*: IEEE; 2006:45–52.
59. Foster I: **Globus toolkit version 4: software for service-oriented systems.** In *Network and Parallel Computing, Volume 3779 of Lecture Notes in Computer Science*. Edited by Jin H, Reed D, Jiang W: Springer; 2005:2–13. [http://dx.doi.org/10.1007/11577188_2]
60. Liu B, Madduri RK, Sotomayor B, Chard K, Lacinski L, Dave UJ, Li J, Liu C, Foster IT: **Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses.** *J Biomed Inform* 2014, **49**(0):119–133. [http://www.sciencedirect.com/science/article/pii/S1532046414000070]
61. Krabbenhöft HN, Möller S, Bayer D: **Integrating ARC grid middleware with Taverna workflows.** *Bioinformatics* 2008, **24**(9):1221–1222. [http://bioinformatics.oxfordjournals.org/content/24/9/1221.abstract]
62. Pronk S, Páll S, Schulz R, Larsson P, Bjelkmar P, Apostolov R, Shirts MR, Smith JC, Kasson PM, van der Spoel D, Hess B, Lindahl E: **GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit.** *Bioinformatics (Oxford, England)* 2013, **29**(7):845–854. [http://www.ncbi.nlm.nih.gov/pubmed/23407358]
63. Noel JK, Whitford PC, Sanbonmatsu KY, Onuchic JN: **SMOG@ctbp simplified deployment of structure-based models in GROMACS.** *Nucleic Acids Res* 2010, **38:**W657–W661. [http://dx.doi.org/10.1093/nar/gkq498]
64. **Jmol: an open-source Java viewer for chemical structures in 3D.** [http://jmol.sourceforge.net/] Accessed 2014-08-07.
65. Prlić A, Yates A, Bliven SE, Rose PW, Jacobsen J, Troshin PV, Chapman M, Gao J, Koh CH, Foisy S, Holland R, Rimša G, Heuer ML, Brandstätter-Müller H, Bourne PE, Willis S: **BioJava: an open-source framework for bioinformatics in 2012.** *Bioinformatics* 2012, **28:**2693–2695.
66. Schuller B, Demuth B, Mix H, Rasch K, Romberg M, Sild S, Maran U, Bala P, del Grosso E, Casalegno M, Piclin N, Pintore M, Sudholt W, Baldridge K: **Chemomentum - UNICORE 6 based infrastructure for complex applications in science and technology.** In *Euro-Par 2007 Workshops: Parallel Processing, Volume 4854 of Lecture Notes in Computer Science*. Edited by Boug'e L, Forsell M, Träff J, Streit A, Ziegler W, Alexander M, Childs S: Springer; 2008:82–93. [http://dx.doi.org/10.1007/978-3-540-78474-6_12]

doi:10.1186/1471-2105-15-292
Cite this article as: Lutz *et al.*: Native structure-based modeling and simulation of biomolecular systems per mouse click. *BMC Bioinformatics* 2014 **15**:292.