jSimMacs for GROMACS: A Java Application for Advanced Molecular Dynamics Simulations with Remote Access Capability

Sanjit Roopra,*,+,† Bernhard Knapp,+,‡ Ulrich Omasits,‡ and Wolfgang Schreiner‡

Department of Bioinformatics, Upper Austrian University of Applied Sciences, Softwarepark 11 4232 Hagenberg, Austria, and Unit for Medical Statistics and Informatics, Section for Biomedical Computer Simulation and Bioinformatics, Medical University of Vienna, Spitalgasse 23 1090 Vienna, Austria

Received May 7, 2009

Molecular dynamics (MD) is a technique to simulate movements of molecular structures to understand their functional behavior. GROMACS is a software package primarily developed for biological MD and offers a huge amount of possible options and settings for tailoring the simulations. This makes it powerful but also complicated to handle. We introduce jSimMacs, a Java application for creating molecular dynamics projects in GROMACS. It simplifies the handling of files and options via an intuitive user interface. Users unexperienced in MD can work along prepared lines, while experts may enjoy a significant relief from the tedium of typing and scripting. Furthermore, jSimMacs supports 3D interactivity and the launch of remote projects on other computers accessible via networks. Thus, jSimMacs not only opens GROMACS to a broader public but also eases the burden of performing series of MD runs, as necessary in parameter studies.

INTRODUCTION

Molecular dynamics (MD) is a method that solves Newton's equations of motion over a specified period of time. Over the last few decades it has become a useful technique with a broad range of applications in biology.[1−4] Via MD structural changes in proteins,[5−7] nucleic acids (DNA and RNA)[8,9] and lipids[10,11] can be investigated. MD is also used for the refinement of structures determined with NMR and X-ray.[2] New areas of application include protein folding, with large scale projects like Folding@home,[12] and free-energy calculations.[13] Current simulations can even cope with large scale phenomena such as a model representing the immune synapse.[6]

There are various MD software packages available, covering different platforms and capabilities (from single core computer to supercomputers, e.g. the BlueGene[14]). Probably most widely known and particularly suitable for biomolecules are CHARMM[15] (supported by CHARMM-GUI[16] and CHARMMing[17]), Amber[18] (supported by Glycam Biomolecule Builder[19]), NAMD[20] (supported by NAMD GUI[21]), GROMOS,[22] and GROMACS.[23] GROMACS is of particular interest because it is distributed under the GPL and can be downloaded for free. It runs on different platforms and exploits special modern features of processor units (e.g., SSE and MMX).[24]

For using Gromacs it is necessary to know the collection of programs (over 75)[24] and libraries, the command line handling under UNIX and the various files generated as output. This may be tough for newcomers to Gromacs even if they are familiar with MD. Therefore, a tool providing easy access and handling can be of great help.

Many different features have to be specified to define a particular simulation environment (force field, water model, initial configuration, length of energy minimization, etc.). Environments may be similar and reused for molecules comparable in type, such as peptide/MHC complexes.[25] Unfortunately, however, GROMACS does not allow saving simulation settings. Many users retreat to files (containing settings and single hardcoded commands) or scripts (launching the appropriate stacks of commands). Such scripts are often specific to particular machines and can not be easily shared between working groups.

Therefore, a user-friendly front-end is desirable and has potential to enlarge the user community for GROMACS. In the following, we briefly outline GUIs already available.

GROMACS itself already includes a simple GUI, which can be started with the Motif or lesstif library.[24] It shows every command in a simple window with boxes to select input, output files, and check various options. However, it cannot guide the user through the workflow and management of a simulation. Other GUIs for Gromacs are GUIMACS[26] and GROMACS GUI.[27]

GUIMACS is implemented in Java and able to create run parameter files, program vault, and an analyzer to start simulations. It is in need of a third party extension called Tcl (Tool Command Language), and it also lacks a simple file manager. Therefore the user can create and start MD projects with GUIMACS, but one has to use a different program to view and manage the resulting files.

GROMACS GUI is implemented in C++ and can only be compiled on Linux systems. It requires several external libraries including Qt4 and Qwt 5. Similar to GUIMACS, it allows the user to go through the simulation steps and it also includes a file management tool.

Since both tools lack certain features, we decided to implement jSimMacs as a GUI for GROMACS. It should run on different platforms and provide the following features:

* Corresponding author e-mail: sanjit.roopra@fh-hagenberg.at.

† Upper Austrian University of Applied Sciences.

‡ Medical University of Vienna.

+ S.R. and B.K. contributed equally to this work

10.1021/ci900248f CCC: $40.75 © 2009 American Chemical Society
Published on Web 10/05/2009

Table 1. Different GROMACS GUIs Compared by Selected Features[a]

	GUIMACS	GROMACS GUI	jSimMacs
remote projects	no	no	yes
3D representation of PDB files	no	no	yes
graph illustration	no	yes	yes
file management	no	yes	yes
integrated help	yes	yes[b]	yes
project management	no	no	yes
storable simulation settings	no	no	yes
screen output	no	yes	yes
file output	no	no	yes
export graphs		pdf	png, svg
MDP writer	yes	yes	yes
GPL	yes	yes	yes
platform independent	yes[c]	no	yes
supports GROMACS version 3.x and 4	no	no	yes

[a] The currently available versions were used from the following locations: GUIMACS 1.0, http://www.gromacs.org/component/option.com_docman/task,doc_details/gid,77/; GROMACS GUI 0.5.2, http://resal.wordpress.com/gromacs-gui/; jSimMacs 0.62b: http://sourceforge.net/projects/jsimmacs/. [b] Needs GROMACS Manual to show help. [c] Using libraries and scripts that are bounded to UNIX systems.

easy to handle, allow to save and reuse simulation settings, include a text editor, and be based on easy manageable, object-oriented code. Additionally, we provide a graphical viewer for molecules and enable to run GROMACS on a remote server via SSH.

jSimMacs addresses both groups of users, beginners and experts. It is completely platform independent and supports above-mentioned features with Gromacs, versions 3.x and 4. A detailed comparison emphasizing the innovations of jSimMacs is shown in Table 1.

MATERIALS AND METHODS

Implementation. To make jSimMacs a simple, reusable, and platform-independent GUI for GROMACS, we implemented it in Java (version 1.5), additionally enabling the idea of remote projects. Most of the features (see Results) could be implemented by using the Java standard library. Some features, namely remote projects, visualization of PDB files, graph visualization, and SVG export, resort to third party libraries (all libraries are implemented in Java and version 1.5 compatible). A summary of the used libraries is shown in Figure 1A.

Remote access is provided via the Trilead SSH2 library,[28] which is supported by Java version 1.4.2, and includes the basic SSH commands, SCP, SFTP, and others. The program automatically detects which kind of authentication is needed. The implementation is realized via a simple logic class, using subclasses (data handlers) to create the correct object (remote or local). The connection details are hidden in a sublayer to render the main program independent. Once connection is established, the program maintains the connection and sends all operations to the server.

To visualize PDB files, the libraries biojava[29] and Jmol[30] are used: biojava is a library that includes file parsers, sequence manipulators, analysis tools and much more, and Jmol is a complete application to visualize structure files (e.g., *.pdb) as 3D images. jSimMacs first uses biojava to read the PDB files, creates an object and then passes it on to Jmol for display. This approach, instead of directly reading by Jmol, leaves important information within jSimMacs, needed for manipulating the molecule later on (see Figure 1B). The structure object of biojava offers detailed information from the PDB file (e.g., chains, number of atoms, sequence). This is used to enable the user to select an amino acid from the sequence data and graphically show the appropriate molecules. In addition, molecules can be picked in the graphical display and will be marked in the sequence data. Since Jmol does not support the structure object, we introduced subclasses and overwrote existing classes of Jmol. Hence for the visualization of structure files jSimMacs works as a translator integrating the flexibility of biojava and Jmol.

Output data of GROMACS (e.g., g_rms) in XVG format are parsed, converted to objects and then graphically displayed using JFreeChart[31] and JCommon.[32] These two libraries already include a lot of features (e.g., zooming and moving the graph, changing the width) and are easy to use. Output in PNG format is directly supported by JFreeChart, whereas SVG files are created using the Batik SVG Toolkit.[33]

Finally, jSimMacs uses the swingutil[34] library that includes templates for graphical user interfaces. This library is of particular interesting for jSimMacs since it provides the graphical representation of tabs implemented in Java 5, thus enabling Java 5 compatibility. Tabs are used for the text viewing and editing part of the program.

Most of the libraries are released under the LGPL, except for the Trilead SSH2 library (BSD style license), the swingutil and Batik libraries (both Apache License). Thus, jSimMacs is released under the GPLv3. Finally, we want to mention that jSimMacs is implemented in a three-tier architecture, using design patterns,[35] and standardized data formats (e.g., XML). This allows programmers to change one layer without interfering with others, making the software easy to modify, maintain, and enhance.

Data. An MD simulation needs different types of data. Most important is the structure file for example taken from the Protein Data Bank (PDB)[36] and is handled in different layers as described above. jSimMacs includes a simple text file viewer to read and manipulate different kinds of text files in various formats necessary for GROMACS. The text layer and the graphical layer are fully interactive with each other, meaning that a change in one of them always entails the corresponding changes in the other layer.

Moreover, jSimMacs includes a complete project management, able to cope with changes made to projects and

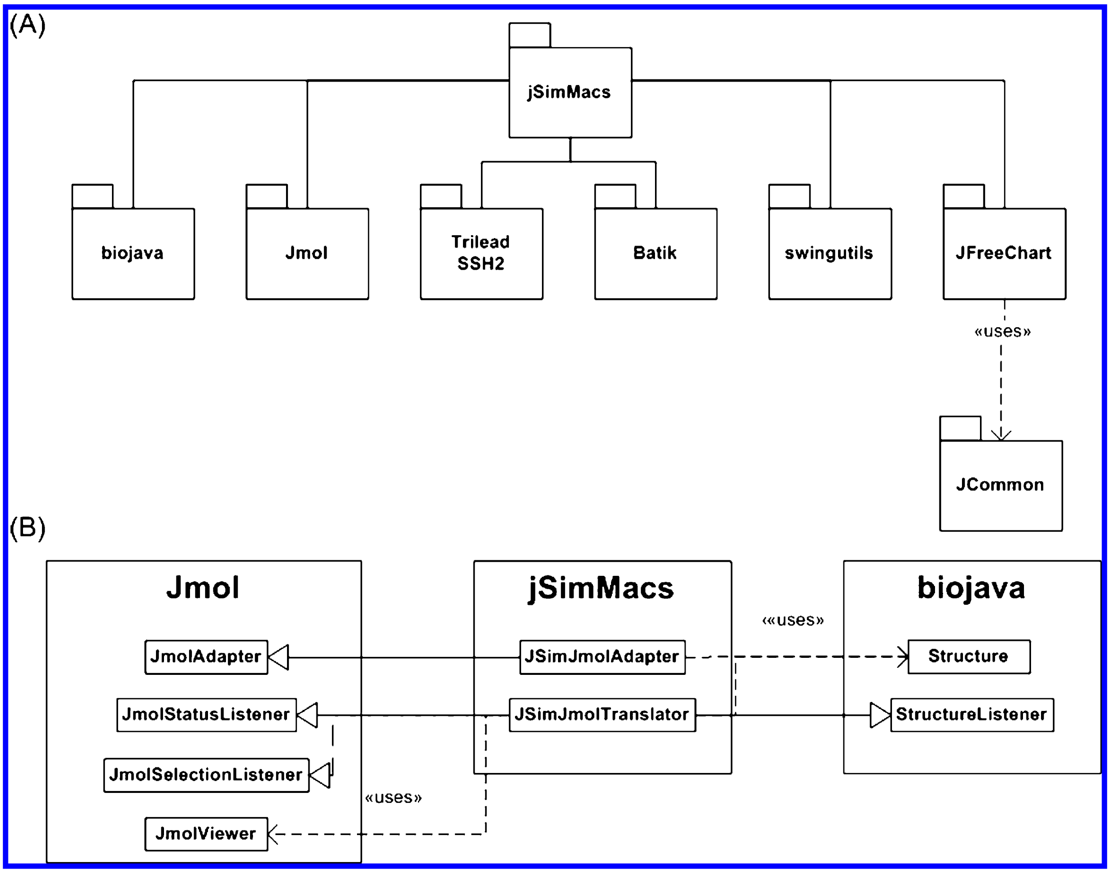


Figure 1. Graph shows the interaction of libraries used in jSimMacs. (A) Libraries used in jSimMacs and their connection to other libraries. (B) A detailed connection graph showing how jSimMacs combines biojava and Jmol objects. Lines with a big arrow are generalizations from classes or interfaces.

simulation runs. This is accomplished by creating an XML file for each project, containing meta-information and simulation settings. This feature enables starting and continuing a simulation at an arbitrary time step, which is of significant practical value. All XML files used in jSimMacs are defined by XML schemas. This makes it easy to use, as only predefined keywords are allowed and the program validates the XML files.

Availability. jSimMacs is available for download at http://sourceforge.net/projects/jsimmacs/ including source code under the GPLv3 license. It requires Java version 5 or higher.

RESULTS

We created a simple to understand GUI for GROMACS, including a lot of features and still straightforward in its workflow.

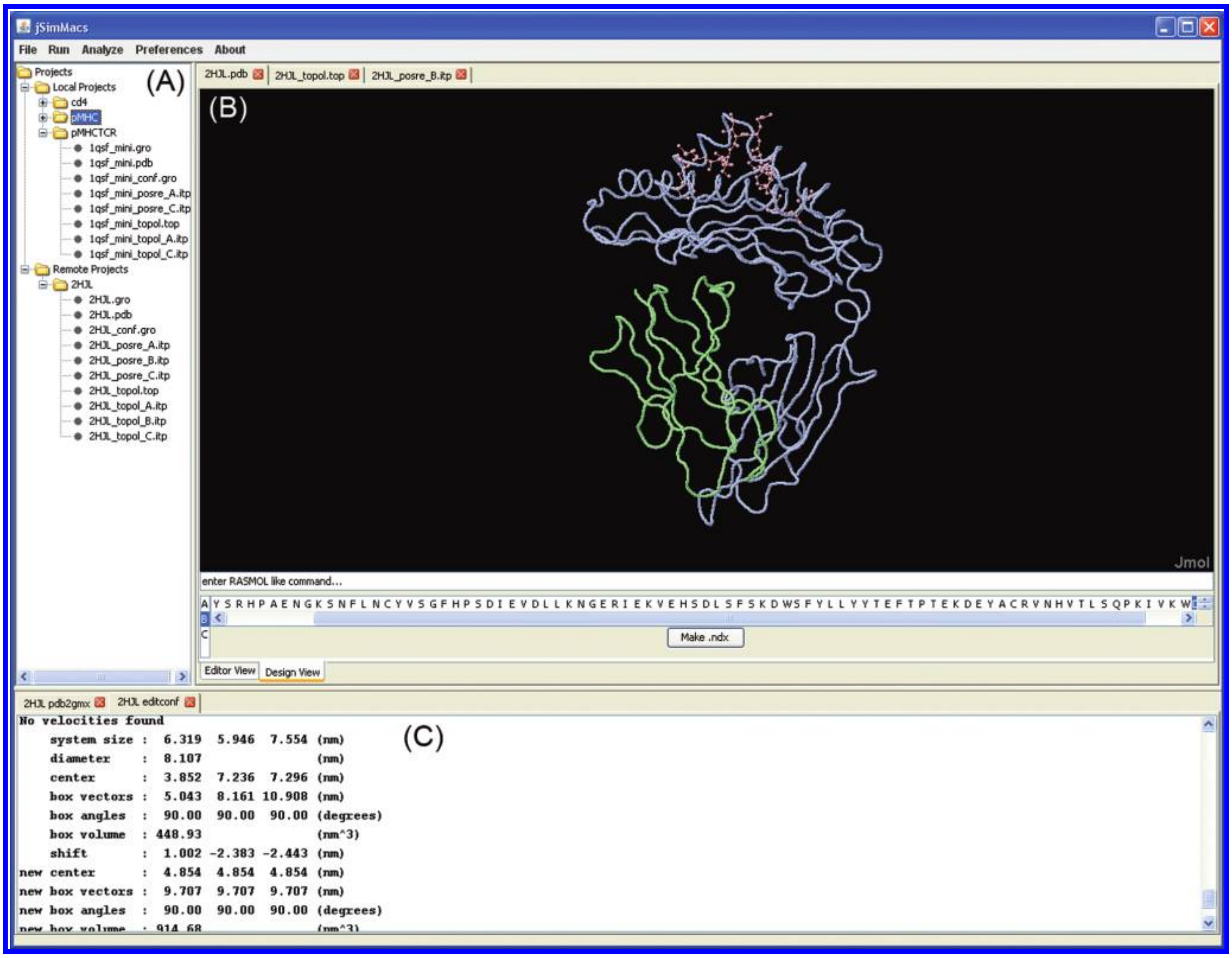
Features. An interactive Jmol display provides graphical interaction for PDB and XVG files. PDB files are visualized as 3D images that can be rotated and manipulated, and parts of the molecule selected by the user. While the selection is specified via the GUI, jSimMacs underneath creates appropriate GROMACS index files, which may then be used, for example, to freeze certain parts of the biomolecules during simulation. Hence it is no more necessary to manually read the PDB files and search for the atoms to be frozen. One selects them interactively via the Jmol interface, see Figure 2.

XVG files are shown as graphs, which can be zoomed, manipulated, and exported to PNG (portable network graphics) or EPS (encapsulated postscript) images. Both PDB and XVG files can also be viewed and manipulated as standard text files with the integrated text editor.

Another feature is the remote capability, allowing to create projects on any computer supporting SSH version 2. Hence, one can run jSimMacs on a desktop Windows platform without GROMACS installed and, thereby, manage simulations on a remote computer which supports SSH and has GROMACS installed. As compared to a simple SSH client, the user fully benefits from interactive capabilities jSimMacs.

A core feature of jSimMacs is its project management capability. Every simulation is saved as a project which may later be reopened and modified. In a project one can create and delete files and directories, delete GROMACS backup files, and import files from remote or local workstations. All project settings are saved in XML files (see also the Materials and Methods) that can be modified through the jSimMacs interface. Here all necessary options are saved, for example, the chosen MPI library, the number of nodes, and the location, if it is a remote project. In addition, all simulation steps can be saved such that it is possible to rerun or continue simulations.

Apparently, it is also possible to start GROMACS simulations using jSimMacs. The user is guided through a predefined default sequence of steps (stored in an XML-file) or may start from scratch. Every output can be alternatively shown on the screen or printed into a file. The application also offers the possibility to analyze an MD run and resume a simulation when crashed. Currently, jSimMacs supports the following GROMACS programs: pdb2gmx, editconf, genbox, grompp, mdrun, tpbconv, trjcat, g_rms,



- Projects
  - Local Projects
    - cd4
    - pMHC
    - pMHCTCR
      - 1qsf_mini.gro
      - 1qsf_mini.pdb
      - 1qsf_mini_conf.gro
      - 1qsf_mini_posre_A.itp
      - 1qsf_mini_posre_C.itp
      - 1qsf_mini_topol.top
      - 1qsf_mini_topol_A.itp
      - 1qsf_mini_topol_C.itp
  - Remote Projects
    - 2HJL
      - 2HJL.gro
      - 2HJL.pdb
      - 2HJL_conf.gro
      - 2HJL_posre_A.itp
      - 2HJL_posre_B.itp
      - 2HJL_posre_C.itp
      - 2HJL_topol.top
      - 2HJL_topol_A.itp
      - 2HJL_topol_B.itp
      - 2HJL_topol_C.itp

enter RASMOL like command...

```
A Y S R H P A E N G K S N F L N C Y V S G F H P S D I E V D L L K N G E R I E K V E H S D L S F S K D W S F Y L L Y Y T E F T P T E K D E Y A C R V N H V T L S Q P K I V K W
B
C
```

```
No velocities found
    system size :  6.319  5.946  7.554 (nm)
    diameter    :  8.107               (nm)
    center      :  3.852  7.236  7.296 (nm)
    box vectors :  5.043  8.161 10.908 (nm)
    box angles  :  90.00  90.00  90.00 (degrees)
    box volume  : 448.93               (nm^3)
    shift       :  1.002 -2.383 -2.443 (nm)
new center      :  4.854  4.854  4.854 (nm)
new box vectors :  9.707  9.707  9.707 (nm)
new box angles  :  90.00  90.00  90.00 (degrees)
new box volume  : 914.68               (nm^3)
```

Figure 2. Screenshot of the user interface of jSimMacs. (A) Project tree browser, showing affiliated directories and files. (B) Interactive molecular structure display, showing the Jmol interface. At the top one can find tabs to switch between different files opened. (C) GROMACS output section displays the command line output of the executables.

g_rmsf, and g_energy. Because of its object oriented design, jSimMacs is easily extendable and maintainable for other GROMACS programs. While jSimMacs is intuitive and easy to use, it also includes an integrated help and tool tips, providing information directly from the GROMACS manual.

Workflow. The basic user-actions needed for creating, running, and analyzing a simulation can be illustrated in a simple workflow (Figure 3). The run and analysis portions are handled by a special wizard that loads all the necessary data, saves the progress, and shows help for each GROMACS command.

DISCUSSION

GROMACS is a powerful tool but also complicated to handle. The workflow to start a MD simulation can be automated with scripts in Perl or Python. However, these small programs need the attention of people familiar with source code and have to be individualized for each particular machine and version of GROMACS. Often such scripts neither meet user expectations regarding convenience nor can they fully support the management of created data and the meaningful assignment of names for the simulation files.

For these reasons we developed jSimMacs, which guides the user through the basic workflow of GROMACS (version 3.x and 4) and thereby significantly helps to manage simulations. The software is licensed under the GPL and has been designed in an object oriented way along design patterns,[33] enabling other users to contribute to the development. Furthermore it includes various helpful features: We implemented a graphical view for certain file types such as PDB and XVG files. This enables the user to have a look at the molecular structure or graph (called design view in jSimMacs) and view the content of the corresponding ASCII-file (called editor view). Changes to the files have direct influence on the graphical views and vice versa.

As a lot of computer work is performed on servers and clusters, we included remote access capability via SSH. This allows users to work on a graphical interface while calculations and analyses run on a more powerful remote machine. Finally, one of the most important features is the project management facility, allowing the user to run his project at any time and stop it whenever he wants, since all settings are saved.

To our knowledge, the PDB graphical viewer and the remote connection via SSH have not been implemented in any other graphical user interface for GROMACS, substantiating the novelty of jSimMacs.

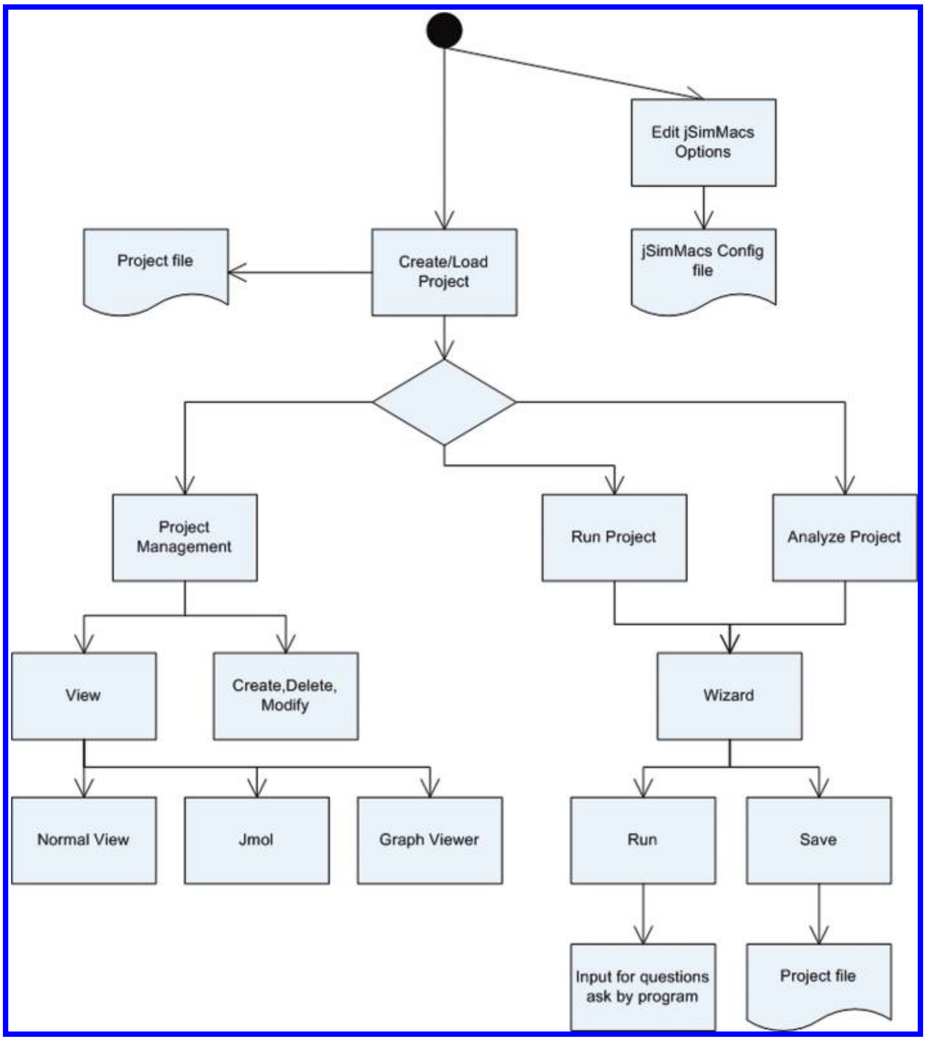


Figure 3. Typical workflow of a jSimMacs simulation. The black circle represents the start, a rectangle a process, and the rhombus a decision state which can be passed multiple times.

The features of jSimMacs will help inexperienced users to carry out MD simulations and will hopefully advance MD simulations to become a more frequent used tool in the chemical laboratories. The authors are aware of the problem that inexperienced users may fall a victim to certain pitfalls of procedure. Therefore several logical checks have already been built in as safeguards, with the perspective of future extension into an expert system. There are frequent discussions if intuitive GUIs can actually be of help for users experienced in command line handling. We like to point out that the automatic handling of files and naming conventions provides a significant benefit also for this group of users. In addition, they of course benefit from the interactive three-dimensional representations and manipulation capabilities, definitely beyond the scope of command line interaction. Therefore we think that jSimMacs is of value for both, experienced and inexperienced users.

CONCLUSION

We have developed jSimMacs as a Java application that automates and simplifies most of the processes performed by GROMACS. Its features aim to support both experienced and inexperienced users. In the future, jSimMacs can be further extended, giving users even more control of the simulation process, such as an interactive display with the possibility to manipulate the current simulation. All in all, jSimMacs has the potential to disseminate MD using GROMACS to a wider public.

ACKNOWLEDGMENT

This work is supported in part by the Austrian Grid Project of the Austrian Ministry of Education, Science and Culture (contract no. GZ 4003/2-VI/4c/2004). The authors declare no conflict of interests.

REFERENCES AND NOTES

(1) Hansson, T.; Oostenbrink, C.; van Gunsteren, W. Molecular dynamics simulations. *Curr. Opin. Struct. Biol.* **2002**, *12*, 190–196.

(2) Karplus, M.; McCammon, J. Molecular dynamics simulations of biomolecules. *Nat. Struct. Biol.* **2002**, *9*, 646–652.

(3) Karplus, M.; Petsko, G. Molecular dynamics simulations in biology. *Nature* **1990**, *347*, 631–639.

(4) Karplus, M.; Kuriyan, J. Molecular dynamics and protein function. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 6679–6685.

(5) Elber, R.; Karplus, M. Enhanced sampling in molecular dynamics: Use of the time-dependent Hartree approximation for a simulation of carbon monoxide diffusion through myoglobin. *J. Am. Chem. Soc.* **1990**, *112*, 9161–9175.

(6) Wan, S.; Flower, D.; Coveney, P. Toward an atomistic understanding of the immune synapse: Large-scale molecular dynamics simulation of a membrane-embedded TCR-pMHC-CD4 complex. *Mol. Immunol.* **2008**, *45*, 1221–1230.

(7) Knapp, B.; Omasits, U.; Bohle, B.; Maillere, B.; Ebner, C.; Schreiner, W.; Jahn-Schmid, B. 3-Layer-based analysis of peptide-MHC interaction: In silico prediction, peptide binding affinity and T cell activation in a relevant allergen-specific model. *Mol. Immunol.* **2009**, *46*, 1839−1844.

(8) Miller, J.; Kollman, P. Theoretical studies of an exceptionally stable RNA tetraloop: Observation of convergence from an incorrect NMR structure to the correct one using unrestrained molecular dynamics. *J. Mol. Biol.* **1997**, *270*, 436–450.

(9) Luo, J.; Bruice, T. Nanosecond molecular dynamics of hybrid triplex and duplex of polycation deoxyribonucleic guanidine strands with a complimentary DNA strand. *J. Am. Chem. Soc.* **1998**, *120*, 1115–1123.

(10) Kandt, C.; Ash, W.; Peter Tieleman, D. Setting up and running molecular dynamics simulations of membrane proteins. *Methods* **2007**, *41*, 475–488.

(11) Heller, H.; Schaefer, M.; Schulten, K. Molecular dynamics simulation of a bilayer of 200 lipids in the gel and in the liquid crystal phase. *J. Phys. Chem.* **1993**, *97*, 8343–8360.

(12) Shirts, M.; Pande, V. S. COMPUTING: Screen savers of the world unite! *Science* **2000**, *290*, 1903–1904.

(13) Simonson, T.; Archontis, G.; Karplus, M. Free energy simulations come of age: Protein−ligand recognition. *Acc. Chem. Res.* **2002**, *35*, 430–437.

(14) Suits, F.; Pitman, M.; Pitera, J.; Swope, W.; Germain, R. Overview of molecular dynamics techniques and early scientific results from the Blue Gene project. *IBM J. Res. Dev.* **2005**, *49*, 475–487.

(15) Brooks, B.; Bruccoleri, R.; Olafson, B.; States, D.; Swaminathan, S.; Karplus, M. CHARMM−A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4*, 187–217.

(16) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A web-based graphical user interface for CHARMM. *J. Comput. Chem.* **2008**, *29*, 1859–1865.

(17) Miller, B. T.; Singh, R. P.; Klauda, J. B.; Hodoscek, M.; Brooks, B. R.; Woodcock, H. L. CHARMMing: A new, flexible web portal for CHARMM. *J. Chem. Inf. Model* **2008**, *48*, 1920–1929.

(18) Case, D.; Cheatham, T., III; Darden, T.; Gohlke, H.; Luo, R.; Merz, K., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(19) Glycam Biomolecule Builder. http://www.glycam.com/CCRC/biombuilder/biomb_index.jsp (accessed Aug 26, 2009).

(20) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

(21) NAMD GUI. http://www.ks.uiuc.edu/Research/vmd/plugins/namdgui/ (accessed Jul 7, 2009).

(22) Christen, M.; Hünenberger, P.; Bakowies, D.; Baron, R.; Bürgi, R.; Geerke, D.; Heinz, T.; Kastenholz, M.; Kräutler, V.; Oostenbrink, C.; Peter, C.; Trzesniak, D.; Van Gunsteren, W. The GROMOS software for biomolecular simulation: GROMOS05. *J. Comput. Chem.* **2005**, *26*, 1719–1751.

(23) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory. Comput.* **2008**, *4*, 435–447.

(24) Lindahl, E.; Hess, B.; van der Spoel, D. GROMACS 3.0: A package for molecular simulation and trajectory analysis. *J. Mol. Model.* **2001**, *7*, 306–317.

(25) Omasits, U.; Knapp, B.; Neumann, M.; Steinhauser, O.; Stockinger, H.; Kobler, R.; Schreiner, W. Analysis of key parameters for molecular dynamics of pMHC molecules. *Mol. Simul.* **2008**, *34*, 781–793.

(26) Kota, P. GUIMACS−A Java based front end for GROMACS. *In Silico Biol* **2007**, *7*, 95–99.

(27) Gromacs GUI. http://resal.wordpress.com/gromacs-gui/ (accessed Apr 8, 2009).

(28) Trilead SSH for Java. http://www.trilead.com/Products/Trilead_SSH_for_Java/ (accessed Apr 8, 2009).

(29) Holland, R. C. G.; Down, T. A.; Pocock, M.; Prlic, A.; Huen, D.; James, K.; Foisy, S.; Drager, A.; Yates, A.; Heuer, M.; Schreiber, M. J. BioJava: an open-source framework for bioinformatics. *Bioinformatics* **2008**, *24*, 2096–2097.

(30) Jmol Java viewer for chemical structures. http://jmol.sourceforge.net/ (accessed Apr 8, 2009).

(31) JFreeChart. http://www.jfree.org/jfreechart/ (accessed Apr 8, 2009).

(32) JCommon. http://www.jfree.org/jcommon/ (accessed Apr 8, 2009).

(33) Batik SVG Toolkit. http://xmlgraphics.apache.org/batik/ (accessed Apr 8, 2009).

(34) Swing Utility Classes. https://swingutil.dev.java.net/ (accessed Apr 8, 2009).

(35) Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison-Wesley Longman: Amsterdam, 1995.

(36) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.

CI900248F