Published in final edited form as:
J Comput Chem. 2013 September 30; 34(25): 2190–2196.

GTKDynamo: a PyMOL plug-in for QC/MM hybrid potential simulations

José Fernando R. Bachega[a], **Luís Fernando S.M. Timmers**[b], **Lucas Assirati**[a], **Leonardo R. Bachega**[c], **Martin J. Field**[d], and **Troy Wymore**[e]

[a]Centro de Biotecnologia Molecular Estrutural, Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos – SP, CEP 13566-590, Brazil

[b]Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas (LABIO) - Programa de Pós-Graduação em Biologia Celular e Molecular (PPGBCM) - Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre - RS, Brazil

[c]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA

[d]DYNAMO/DYNAMOP, Institut de Biologie Structurale – Jean-Pierre Ebel, 41 rue Jules Horowitz, 38027 Grenoble Cedex 1, France (UMR 5075 – CEA/CNRS/UJF Grenoble I)

[e]Pittsburgh Supercomputing Center, Pittsburgh, USA

Abstract

Hybrid quantum chemical (QC)/molecular mechanical (MM) potentials are very powerful tools for molecular simulation. They are especially useful for studying processes in condensed phase systems, such as chemical reactions, that involve a relatively localized change in electronic structure and where the surrounding environment contributes to these changes but can be represented with more computationally efficient functional forms. Despite their utility, however, these potentials are not always straightforward to apply since the extent of significant electronic structure changes occurring in the condensed phase process may not be intuitively obvious. To facilitate their use we have developed an open-source graphical plug-in, GTKDynamo, that links the PyMOL visualization program and the pDynamo QC/MM simulation library. This article describes the implementation of GTKDynamo and its capabilities and illustrates its application to QC/MM simulations.

Keywords

Graphical plug-in; hybrid quantum chemical / molecular mechanical potentials; molecular simulation; pDynamo molecular modeling package; PyMOL molecular visualization package; Python scripting language

*ferbachega@gmail.com

Introduction

The molecular simulation of condensed-phase chemical reactions is a very challenging computational task. Quantum chemical (QC) approaches provide the most realistic description of such systems because they can determine molecular electronic structure from first principles and so permit the exploration of processes such as bond-breaking and forming, charge transfer and electronic excitation. QC methods are, however, computationally expensive and are limited to relatively small systems of several hundred atoms or less, depending upon the method that is being employed. Alternatives to QC approaches are based on molecular mechanical (MM) potentials or force-fields. These can study systems of almost arbitrary size and are widely employed. In the area of biomacromolecular simulation, for example, they are used for structure refinement and prediction, the study of macromolecular dynamics and of protein-ligand interactions. Unfortunately, MM potentials suffer from limitations of their own, including their dependence on a large number of empirical parameters and their unsuitability, in many cases, for studying processes where electronic structure rearrangements occur, such as chemical reactions.

For many condensed-phase chemical reaction simulations, hybrid approaches that combine both QC and MM descriptions of a system are appropriate. Warshel and Levitt first introduced a method of this type in their study of the reaction catalyzed by the enzyme lysozyme[1]. Since then, QC/MM potentials have become a standard tool for molecular simulation and dominate for some applications, such as in the investigation of the mechanisms of condensed-phase reactions for which a realistic description of the environment is required, as in enzymes. There are a wide variety of QC/MM methods and their development and application have been extensively documented[2-6].

Many molecular modeling packages now contain QC/MM capabilities. One of the earliest that was developed with these calculations in mind, however, was the Dynamo series of programs. The first publicly available version, named fDynamo, was written in Fortran 90/95 and was released under an open-source license in 1999[7-8]. fDynamo was superseded in 2008 by a new version of the library, pDynamo, that was written in the scripting language Python and in the compiled language C[9-10]. pDynamo, is much more capable than fDynamo as it is compatible with most of the common biomolecular force-fields, has a wide range of semi-empirical and *ab initio* QC methods, and a large number of simulation algorithms. In addition, its use of Python makes it much easier to use and more powerful than fDynamo and other competing packages.

The construction of models for hybrid potential simulations is often challenging since reasonable boundaries between the QC and MM regions need to be identified. A graphical interface to pDynamo, named GTKDynamo, was developed which links pDynamo and the PyMOL program, and facilitates this and other critical model building tasks as well as the analysis of pDynamo output. PyMOL is a very popular and powerful visualization tool that is specialized for, but not limited to, the handling of biological macromolecules. It is written largely in Python, which facilitates the creation of third-party plug-ins, and also makes it a

natural fit for pDynamo. Examples from among the many PyMOL plug-ins include: Autodock/AutodockVina[11], APBS[12], CAVER[13] and PocketPicker[14].

In what follows, we describe our implementation of the GTKDynamo plug-in and some of its capabilities. GTKDynamo itself is distributed under an open-source license and is available at sites.google.com/site/gtkdynamo.

Program Overview and Implementation

In this article we present a graphical plug-in, called GTKDynamo, for the PyMOL program that provides an interface to the pDynamo molecular modeling library. GTKDynamo is implemented as an extra PyMOL window that permits access to a wide range of pDynamo's features, including the import and construction of molecular systems, the import and creation of atom selections, the editing of coordinates, trajectory analysis, single-point energy calculations, geometry optimization, molecular dynamics simulations, normal mode analysis, reaction coordinate scans and potential of mean force calculations. In addition to pDynamo, GTKDynamo also has an extensions window that connects to other third-party programs with complementary functionality.

GTKDynamo is written in the Python scripting language and employs the GTK+ toolkit. The latter is compatible with the TK toolkit used in PyMOL but has more sophisticated widgets and gives a friendlier interface. Extensive use is also made of the Matplotlib toolkit for plotting the results generated by pDynamo programs. A snapshot of a typical GTKDynamo session is shown in Figure 1.

The approach adopted by GTKDynamo to connect pDynamo and PyMOL is similar to that used by Seeliger *et al.*[11] in their AutoDock plug-in. All communication between the two programs is indirect, using files. Coordinate files in MOL2, PDB and XYZ formats are employed for molecular structures, whereas Gaussian cube files are used for surfaces, such as the electron density, the electrostatic potential and molecular orbitals. All files generated by GTKDynamo are saved in a directory designated by the user. Figure 2 shows a flow chart illustrating how GTKDynamo interacts with the other programs and libraries.

Model Preparation

GTKDynamo facilitates greatly the preparation of a system for simulation, which is one of the most important and delicate operations in molecular modeling. A straightforward way to load a system into GTKDynamo is to use a coordinate file. This can be done via pDynamo using one of the many formats that it recognizes or, alternatively, by importing coordinates from one of the currently-defined PyMOL objects. Likewise, it is possible to export the coordinates of a system from GTKDynamo to PyMOL where they can be visualized and otherwise manipulated using PyMOL's interactive facilities.

To calculate an energy for a system once it is available in GTKDynamo, it is necessary to associate an “energy model” to the system. The easiest to employ is a QC model as in this case the information in a coordinate file is often sufficient. Of course, pure QC calculations are only appropriate for relatively small systems due to computational expense.

GTKDynamo's main window contains a section in which the user can define the necessary attributes of the QC description. These include the electronic state of the system (charge and multiplicity) along with the QC method to employ. Currently available are a number of semi-empirical (AM1, PM3, PM6, RM1) and DFT methods that are natively implemented in pDynamo, together with access to the interface that pDynamo possesses to the ORCA *ab initio* QC program[15]. If the latter is chosen, a new window appears with which ORCA-related options can be specified.

The construction of an MM model is more complicated than for a QC model as more information beyond element type and Cartesian coordinates are required. pDynamo has a comprehensive set of python utilities for building models with a variety of force-fields, but GTKDynamo does not currently have an interface to these, although this will be implemented in the future. In the meantime, an MM model can be defined in GTKDynamo by reading files that have been generated by other programs and so already possess this information. These include pDynamo's native serialization formats (pkl and yaml), as well as the third-party formats employed by the Amber[16] (top/crd), CHARMM[17] (psf) and Gromacs[18] (top/gro) packages. CHARMM psf files are also employed by the NAMD, PSFGEN, VMD and X-PLOR programs. Figure 3 illustrates the import of a system, in this case the enzyme triose phosphate isomerase, using files in Amber format.

Once a system has been defined, GTKDynamo can manipulate it in a number of ways. Some of the principal manipulations involve the use of selections which can be defined interactively by picking atoms in PyMOL and then importing the selection object into GTKDynamo. Two important operations that use selections are “atom fixing” and “pruning”, both of which can reduce the cost of particular types of simulation. Atom fixing freezes the positions of the selected atoms so they do not move in a geometry optimization or MD simulation, whereas pruning creates a system of reduced size by deleting selected atoms. An example of pruning using GTKDynamo is given in Figure 4.

A third important use of atom selections is in the definition of the QC region for a hybrid potential QC/MM calculation. The procedure for doing this is illustrated in Figure 5. It is necessary to start off with a system for which an MM model has been defined. Users can then create a selection specifying the atoms that are to be treated quantum chemically. This selection may then be imported into GTKDynamo, and the QC model for the QC region can be set using the same procedure as for a pure QC calculation. Finally, once all parameters have been defined, the QC atoms are automatically highlighted in the PyMOL structure window using a ball and stick representation, which is readily visualized.

GTKDynamo can write files in pDynamo's native serialization formats (pkl and yaml) as well as reading them. This is useful, because such files can be employed to store all the information about a system (MM, QC, fixed atoms, etc.) once it has been setup, without having to repeat the model preparation process. In addition, GTKDynamo/PyMOL sessions can be saved using GTKDynamo's own project format, gtkdyn. This is a simple Python script that connects a PyMOL pse session file with the appropriate pDynamo serialization files and is particularly useful when many different GTKDynamo and PyMOL objects have been defined.

Simulation Types

GTKDynamo's simulation tools encompass many of the standard calculation types that are found in computational chemistry and molecular simulation. These include:

- Single-point energy calculations. MM, QC and QC/MM potentials can all be used.
- Geometry optimization. Currently only pDynamo's conjugate gradient and steepest descent optimization algorithms are exposed. The interface allows the user to specify the number of optimization cycles to perform, the convergence criterion to employ, and whether a trajectory of structures is to be saved during the calculation.
- Molecular dynamics simulations. These can be performed using three different integration algorithms — velocity Verlet, leap-frog and a stochastic Langevin method. GTKDynamo allows the user to set the values of relevant parameter for these methods, including the time step, the length of the simulation, the temperature and pressure, the frequency of logging and trajectory writing, and the seed for the random number generator.
- Nudged-elastic-band (NEB) reaction path calculations. This algorithm finds a minimum energy path between two structures (reactants and products). In the NEB window of GTKDynamo, the reactant and product structures, along with other relevant parameters, such as the number of structures to represent the path, can all be set. Figure 6 illustrates how a NEB calculation is set up with GTKDynamo and the type of results that are obtained at the end of the calculation.
- Potential energy scans (PES) in either one or two dimensions. Currently scans can be done using distances or a linear weighted combination of distances as constraints, and each point of the scan involves a geometry optimization using either the conjugate gradient or steepest descent algorithms. Scan parameters that can be set are the variable ranges, the number of points along each dimension and the constraint force constants.
- Potential of mean force (PMF) simulations. This is a powerful method that computes the free energy as a function of user-defined reaction coordinate variables. Its implementation in GTKDynamo is similar to that of the PES scan, except that a short MD simulation is performed at each window (equivalent to a scan point), instead of a geometry optimization. The free-energy surface can be reconstructed after simulation using the reaction-coordinate trajectories generated during the simulations and the WHAM method implemented in pDynamo.

Analysis of Results

Calculations run with the pDynamo library can output results in a wide range of formats. In GTKDynamo, all pDynamo processes have been designed to generate a plain-text log file that contains the essential results from a given simulation. GTKDynamo is able to read and interpret these log files and display graphics of the relevant information, which can then be saved and manipulated by the user. This is a convenient way of creating both graphical and structural figures. Processing of log files is done automatically whenever a pDynamo job

initiated from GTKDynamo terminates, but it can also be performed for previously-generated GTKDynamo/pDynamo log files.

Another important class of results file that pDynamo produces are trajectories. These come in a number of formats, both pDynamo-specific and third-party, and can hold a variety of information, including coordinates, energies, reaction-coordinate variables and velocities. GTKDynamo can handle several types of pDynamo trajectory. It permits users to load several trajectories at once and to select which objects on the trajectory are to be treated. In addition, it can pass the loaded data to PyMOL's powerful trajectory manipulation tool with which it is possible to perform a wide range of structural and visual analyses.

Extensions

The “Extensions” section of the GTKDynamo plug-in gives access to programs other than pDynamo. There are currently interfaces to AmberTools and to the GROMACS package. One of the principal reasons for doing this was to provide an alternative means of preparing a fully defined MM model of a system without the need to employ pDynamo directly. The AmberTools interface is the most general as it can construct topologies and Amber force-field parameters for molecules of arbitrary type. This is especially useful for protein systems that contain ligands or other organic molecules. GROMACS can also build MM models, using the Amber or Charmm force-fields, although topologies for “non-standard” molecules need to be provided by the user.

The topologies and parameters built by the AmberTools and GROMACS extensions are fully readable and editable by the pDynamo library as well, of course, by the Amber and GROMACS programs themselves. This provides extra flexibility as it means, for example, that these other programs could be employed to perform initial classical MM/MD simulations of the constructed systems, before switching to pDynamo and the GTKDynamo interface for QC/MM studies. Finally, the “Extensions” section also exposes other capabilities of the interfaced packages in addition to model building. Thus, it is possible to minimize an MM model directly using the GROMACS package. It is intended to add further such options in the future.

Final Considerations

In this article, we have described a PyMOL plug-in interface, GTKDynamo that has been designed for the modeling of reaction pathways in biological systems. The plug-in facilitates system preparation and analysis, and provides a number of simulation options, including single point energy calculations, geometry optimization, PES scans in one and two dimensions with a variety of reaction coordinate variables, NEB reaction path finding and determination of PMF free-energy surfaces. GTKDynamo employs the pDynamo molecular modeling program as its principal computational engine but interfaces to other programs, including AmberTools and GROMACS, are also available. GTKDynamo is issued under an open-source licence and is available for download for Linux and Mac OS-X platforms at sites.google.com/site/gtkdynamo. Extensive documentation and tutorials exist on the website and these have been used and improved through their dissemination at workshops held in Pittsburgh and Barcelona. The use of GTKDynamo for instruction on the proper use of

hybrid QC/MM methods could have a broad impact within the biomolecular simulation community.

System Requirements

GTKDynamo was developed under Linux (Ubuntu version 11.10) with pDynamo version 1.7.2 (www.pdynamo.org) and PyMOL versions 1.4 and 1.5 (www.pymol.org). It has also been successfully installed and operated on a Macintosh machine with MAC OS-X version 10.7.5. GTKDynamo requires version 2.6 or 2.7 of the Python scripting language (www.python.org) and some additional third-party Python packages, including Numpy (numpy.scipy.org), Matplotlib (matplotlib.org) and pyGTK (www.pygtk.org). The "Extensions" section requires the AmberTools package (ambermd.org/#AmberTools) and the GROMACS molecular modeling program (www.gromacs.org) to work properly.

Acknowledgments

The authors acknowledge financial support from the National Center for Research Resources at the National Institute of Health (grant number RR06009) and the CNPq (to JFRB) and CAPES (to LFSMT) for fellowships. They also thank Professors Richard C. Garratt, Rinaldo W. Montalvao and Osmar Norberto de Souza for their insightful comments and discussions.

References

1. Warshel A, Levitt M M. J Mol Biol. 1976; 103(2):227–249. [PubMed: 985660]
2. Senn HM, Thiel W. Curr Opinion Chem Biol. 2007; 11:182–187.
3. Lin ZH, Jasper AW, Truhlar DG. J Am Chem Soc. 2007; 129(48):14899–14910. [PubMed: 17994736]
4. Senn HM, Thiel W. QM/MM Methods for Biomolecular Systems Angewandte Chemie International Edition. 2009; 48:1198–1229.
5. Thiel, W. QM/MM Methodology: Fundamentals, Scope, and Limitations. In: Grotendorst, J.; Attig, N.; Blugel, S.; Marx, D., editors. Multiscale Simulation Methods in Molecular Sciences. Vol. 42. Institute for Advanced Simulation, Forschungszentrum Julich; 2009. p. 203-214.NIC Series2009
6. Lonsdale R, Ranaghan KE, Mulholland AJ. Chem Commun. 2010; 46:2354–2372.
7. Field MJ, Albe M, Bret C, Proust-De-Martin F, Thomas A. J Comput Chem. 2000; 21(12):1088–1100.
8. Field MJ. J Comput Chem. 2002; 23(1):48–58. [PubMed: 11913389]
9. Field, MJ. A Pratical Introduction to the Simulation of Molecular System. Cambridge University Press; Cambridge UK: 2007.
10. Field MJ. J Chem Theor Comp. 2008; 4:1151–1161.
11. Seeliger D, de Groot BL. J Comput-Aided Mol Des. 2010; 24:417–422. [PubMed: 20401516]
12. Baker NA, Sept D, Joseph S, Holst MJ, McCammon JA. Proc Natl Acad Sci USA. 2001; 98:10037–10041. [PubMed: 11517324]
13. Petrek M, Otyepka M, Banás P, Kosinová P, Koca J, Damborský J. BMC Bioinformatics. 2006; 7:316. [PubMed: 16792811]
14. Weisel M, Proschak E, Schneider G. Chem Cent J. 2007; 1:7. [PubMed: 17880740]
15. Neese F. WIREs Comput Mol Sci. 2011; 2:73–78.
16. Case, DA.; Darden, TA.; Cheatham, TE., III; Simmerling, CL.; Wang, J.; Duke, RE.; Luo, R.; Walker, RC.; Zhang, W.; Merz, KM.; Roberts, B.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Kolossvory, I.; Wong, KF.; Paesani, F.; Vanicek, J.; Liu, J.; Wu, X.; Brozell, SR.; Steinbrecher, T.; Gohlke, H.; Cai, Q.; Ye, X.; Wang, MJ.; Hsieh, J.; Cui, G.; Roe, DR.; Mathews, DH.; Seetin, MG.; Sagui, C.; Babin, V.; Luchko, T.; Gusarov, S.; Kovalenko, A.; Kollman, PA. AMBER 11. University of California; San Francisco: 2010.

17. MacKerell AD, Bashford D Jr, Bellott M, Dunbrack RL Jr, Evanseck J, Field MJ, Fischer S, Gao J, Guo H, Ha S, Joseph D, Kuchnir L, Kuczera K, Lau FTK, Mattos C, Michnick S, Ngo T, Nguyen DT, Prodhom B, Reiher IWE, Roux B, Schlenkrich M, Smith J, Stote R, Straub J, Watanabe M, Wiorkiewicz-Kuczera MJ, Yin D, Karplus M. J Phys Chem B. 1998; 102:3586–3616. [PubMed: 24889800]

18. van der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJ. J Chem Theor Comp. 2005; 4:435–447.

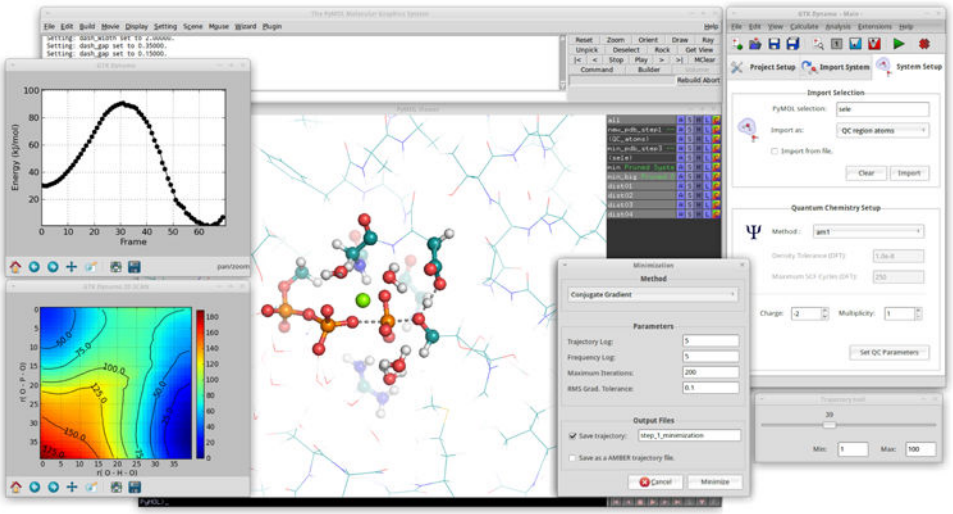

Figure 1.
An image with a snapshot of the GTKDynamo PyMOL interface.

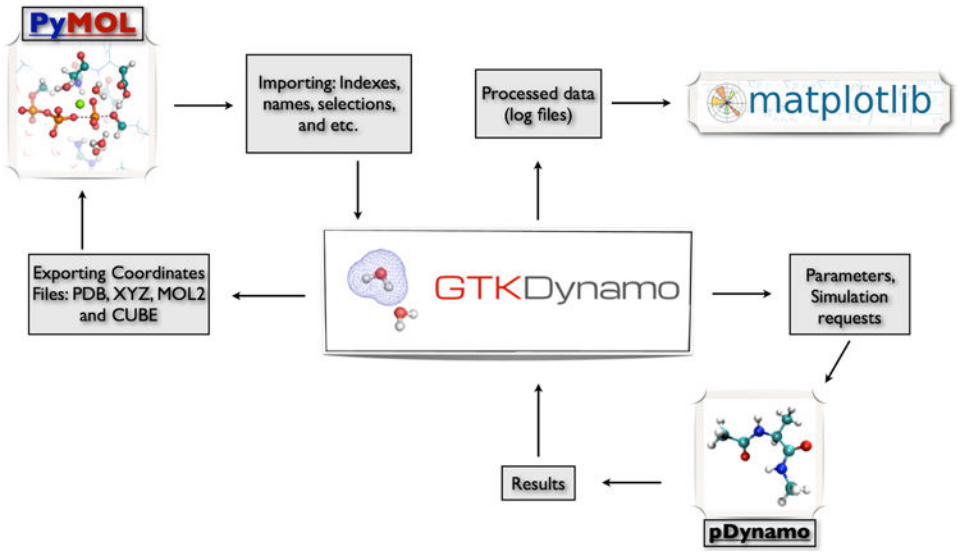

Figure 2.
A flow chart illustrating the interactions between GTKDynamo and the other programs and libraries.

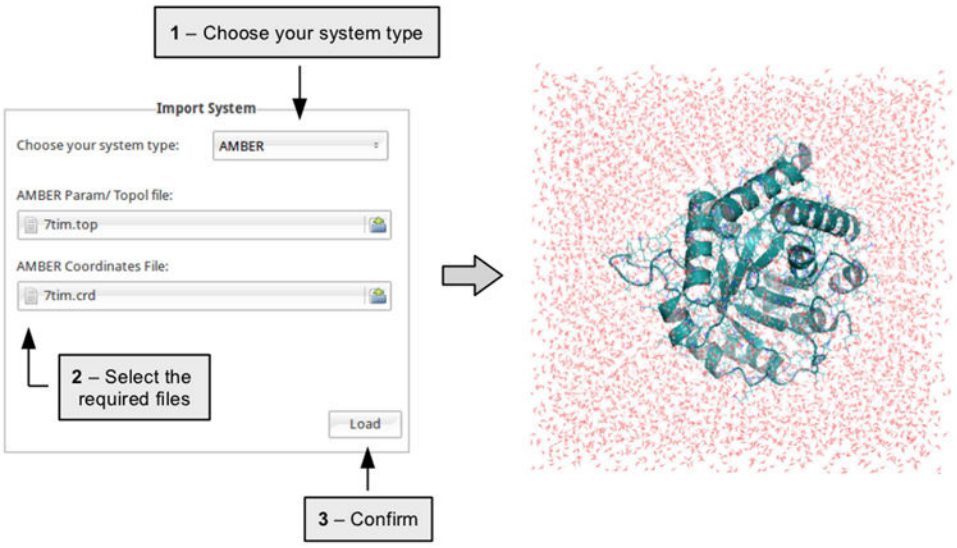

Figure 3.

A schematic showing how a system with a defined MM model can be imported into GTKDynamo. In this case, the files employed are in AMBER format.

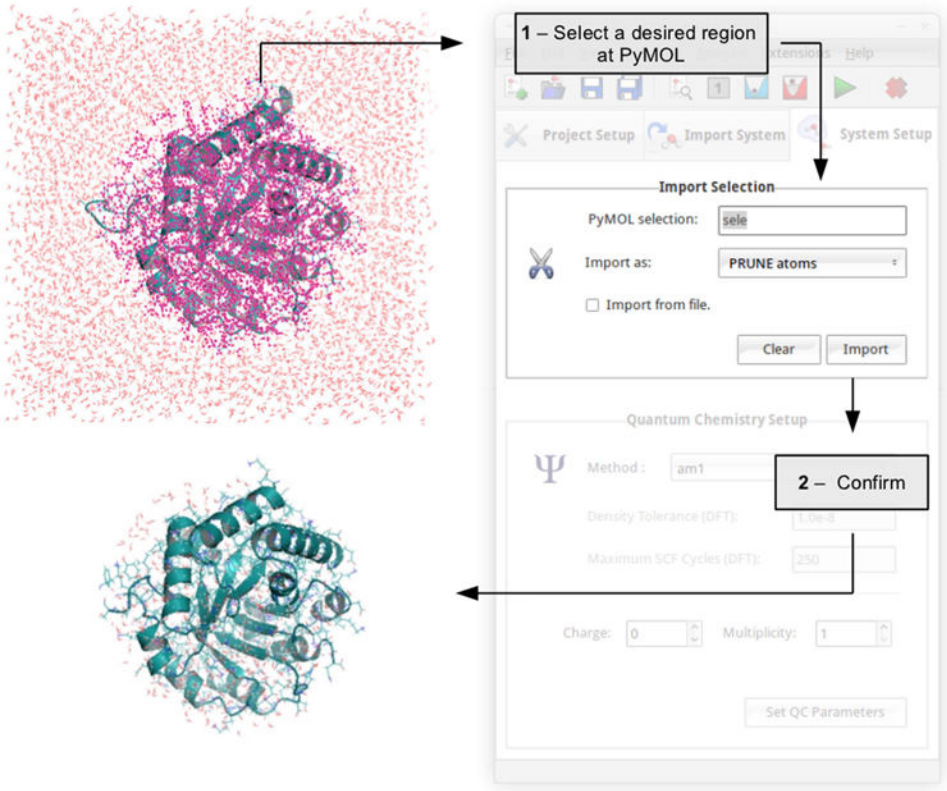

Figure 4.

A schematic showing how a system can be edited. In this case, the system is reduced in size (or “pruned”) to make it more suitable for a hybrid potential simulation.

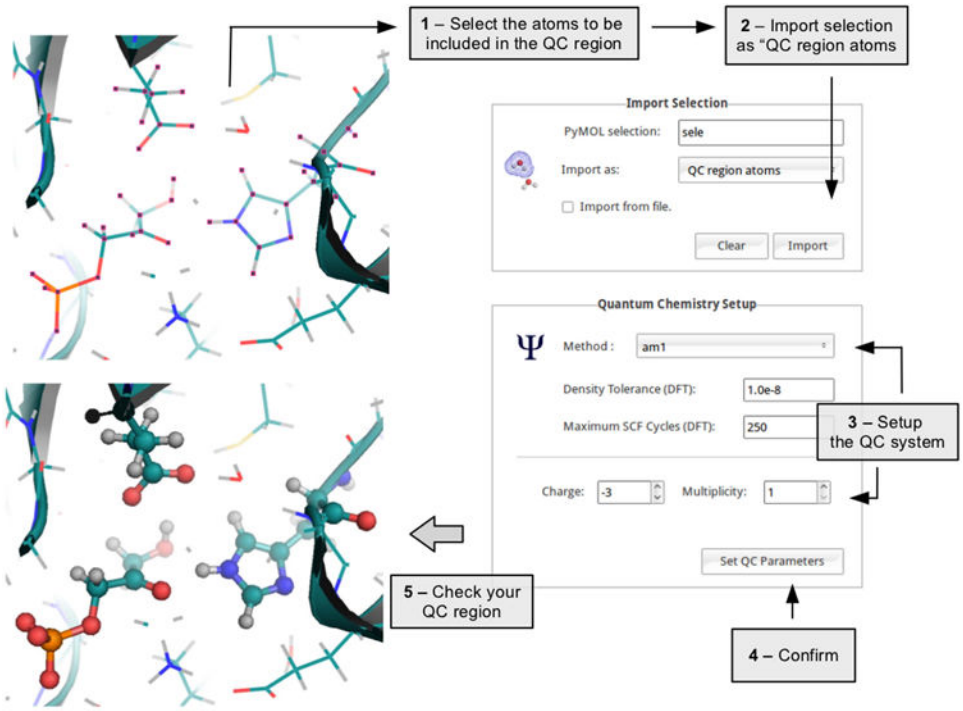

Figure 5.
A schematic showing how the QC region of a protein system is defined using the interface.

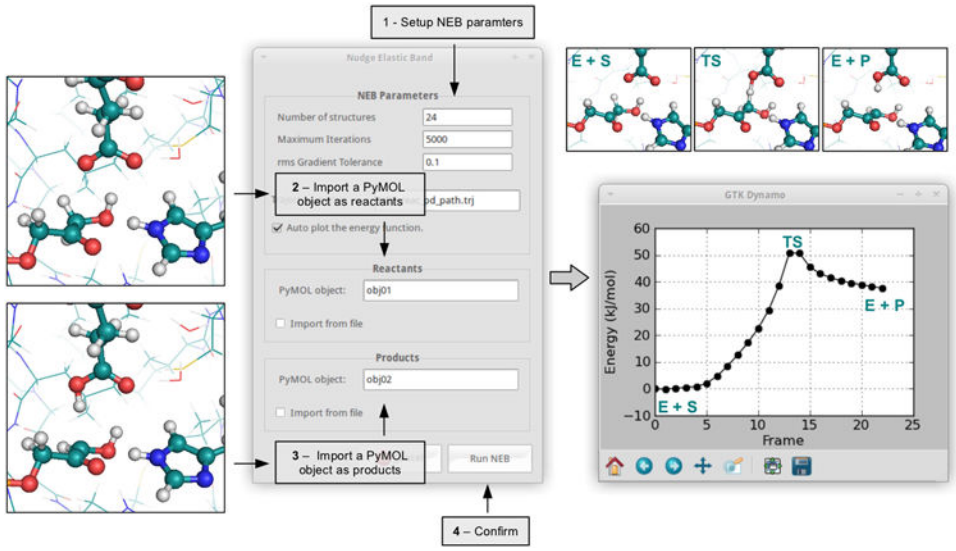

Figure 6.

A schematic showing how a QC/MM NEB calculation is performed with GTKDynamo. The resulting energy profile as a function of structure number is plotted at the end of calculation.