Improving Productivity while managing extensive Molecular Dynamics simulation data

Avimanyu Bandyopadhyay[1]

[1] Department of Biotechnology, Heritage Institute of Technology, Kolkata, West Bengal, India

Corresponding Author:

Avimanyu Bandyopadhyay[1]

Bioinformatics Lab, Department of Biotechnology, Heritage Institute of Technology, Chowbaga Road, Anandapur, PO:East Kolkata Township, Kolkata, West Bengal, 700107, India

Email address: avimanyu.bandyopadhyay@heritageit.edu.in

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

Abstract

Background. This paper discusses the difficulties experienced by bioinformaticians while working with extensive data generated from extended molecular dynamics simulations. For better experimental analysis, it often becomes crucial to conduct simulations up to extended periods of time. When with limited resources, running a complete simulation up to a desired length of time can become quite difficult to be performed at one go. So, a new approach is proposed to simplify handling such data for better productivity.

Methods. Development of a front-end interface.

Results. Showing improvements in usability.

Discussion. Comparing a command line vs graphical interface approach for managing extended molecular dynamics simulations reveal the benefits of the latter by proposed method of use.

Introduction

Molecular dynamics (Marco De Vivo 2016) simulations have become a great deciding factor in drug development in today's time. One such molecular dynamics software for the same is Desmond (Kevin J. Bowers 2006). For better experimental analysis, it often becomes crucial to conduct simulations up to extended periods of time. When with limited resources, running a complete simulation up to a desired length of time can become quite difficult to be performed at one go. The solution to this is to break them up into smaller time intervals and conduct them separately but sequentially. Such simulations can often take many days to complete. Each of these sequential simulations, starting from the end of the first simulation, becomes an extended simulation with the help of checkpoints, which contain information about simulation progress. At the end of the complete full-length simulation, bioinformaticians have to consolidate all of these generated data that gets accumulated. If there are *n* number of extended simulations, there would be *n+1* number of trajectories (Robert T. McGibbon 2015) generated, which contain conformational information of the 3D protein structure that had undergone simulation. In order to view the complete trajectory of the structure starting from 0 unit time until the end of the simulation, it would be essential to combine all of these sequential trajectories into one single and complete trajectory. Only then would it be possible to observe and interpret the corresponding plots pertaining to molecular statistics of the corresponding simulation. Combining all these trajectories is only possible with the help of commands. This can become difficult to manage when a huge number of sequential simulations have been performed. To make this process easier for bioinformaticians, a new approach is proposed.

Materials & Methods

By using *Tkinter*, one of Python's own graphical user interface (GUI) libraries, a front-end interface has been developed to make it a lot more convenient for the user to merge trajectories. The mini-tool named *TrajectoryMergeAssist*, archived and documented on Zenodo (Avimanyu B. 2018), allows the user to easily choose the required files on the GUI and merge them with the help of the command that would quietly run for the user on the back-end. The software is available on GitHub and is Open Source under GPL 3.0.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

System configuration used:

Own System:

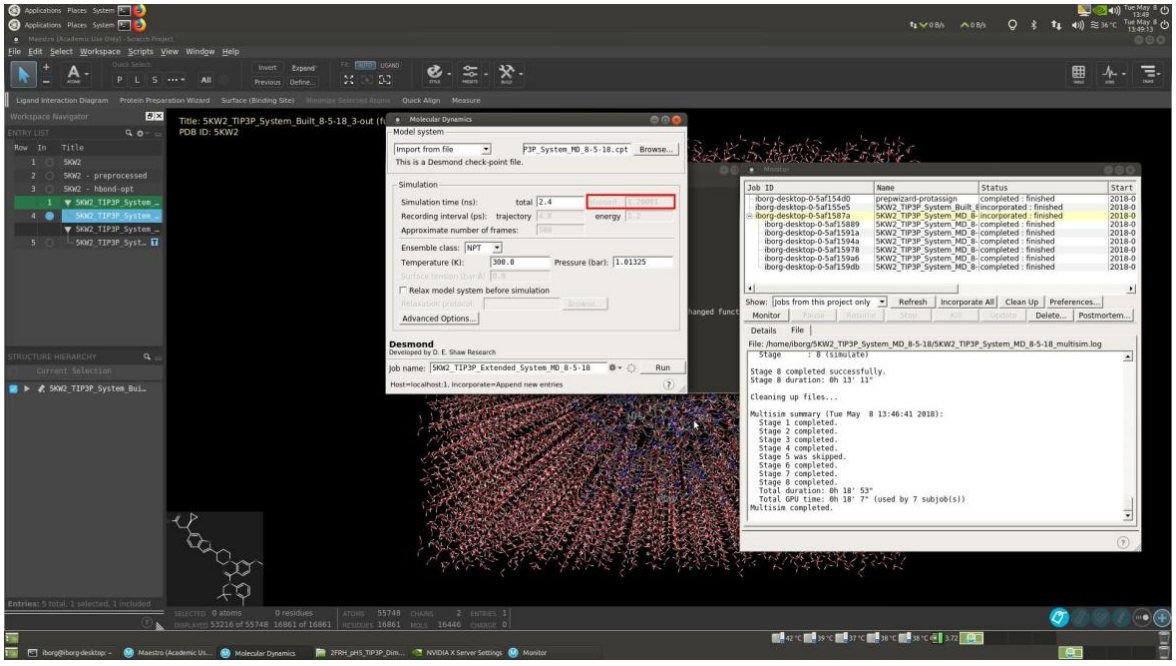
Operating System: Ubuntu Linux 16.04 LTS 64 bit
GPU: Asus NVIDIA Maxwell Titan X 12 GB
CPU: Intel i7 4770K at 3.5 Ghz
256 GB SSD
500 GB + 2 TB HDD
32 GB DDR3 RAM

Lab System:

Operating System: Ubuntu Linux 14.04 LTS 64 bit
GPU: Zotac NVIDIA Kepler GT730 2 GB
CPU: Intel i3 2100 at 3.1 Ghz
500 GB HDD
8 GB DDR3 RAM

The following screenshot shows a simulation being extended for a prolonged molecular dynamics simulation for a protein structure with PDB id 5KW2 built in a TIP3P solvent model on Desmond (D. E. Shaw Research, n.d.) running via Maestro (Schrödinger, n.d.):

Example of extending a simulation up to 2.4 ns that has already run until 1.2 ns on Desmond with a checkpoint (".cpt") file:



PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

Discussion

Current method of merging trajectories:

Commands:

Version 18-1 or later

$SCHRODINGER/run trj_merge.py *out.cms traj1 traj2 -o merged*

Earlier versions

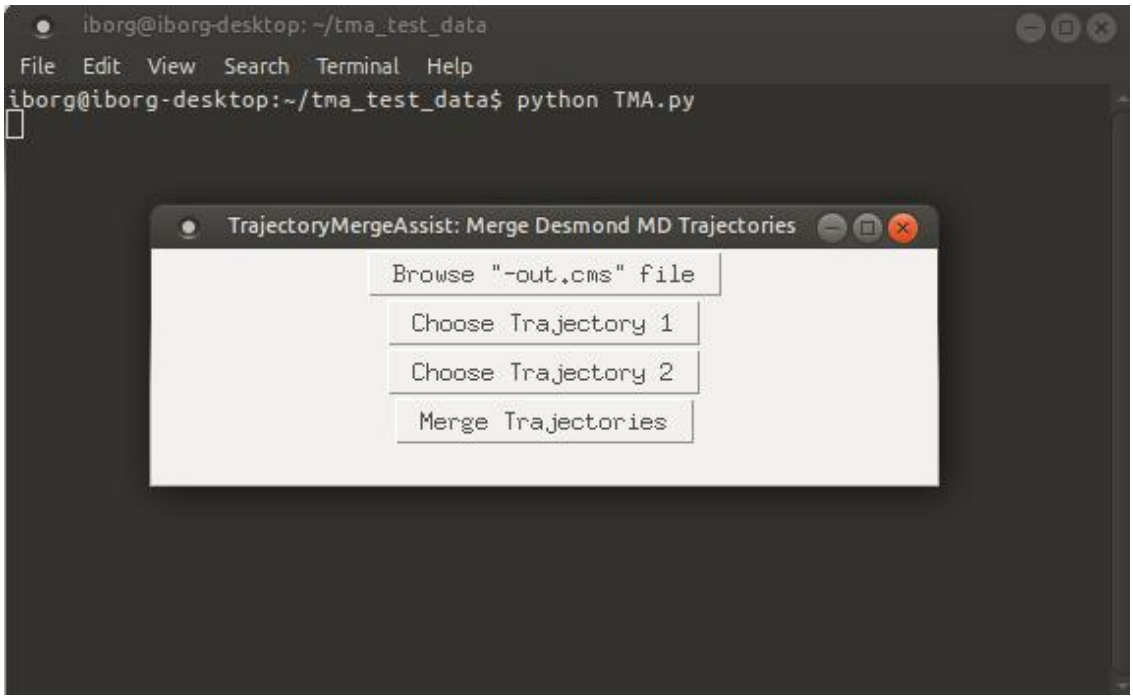
$SCHRODINGER/run -FROM desmond manipulate_trj.py *input*.cms *outfile in-trj1 in-trj2*

Even with a powerful Nvidia Titan X GPU, it approximately takes 20 minutes for a simulation of 1.2 ns to complete. For optimum analysis and interpretations, simulation times can be extended up to 100 ns or even 1000 ns for absolute confirmations of inferences. When it comes to a huge number of extended simulations for such experiments, using commands for merging *n* number of extended trajectories with the initial one can eventually become quite inconvenient.

Proposed method of merging trajectories:

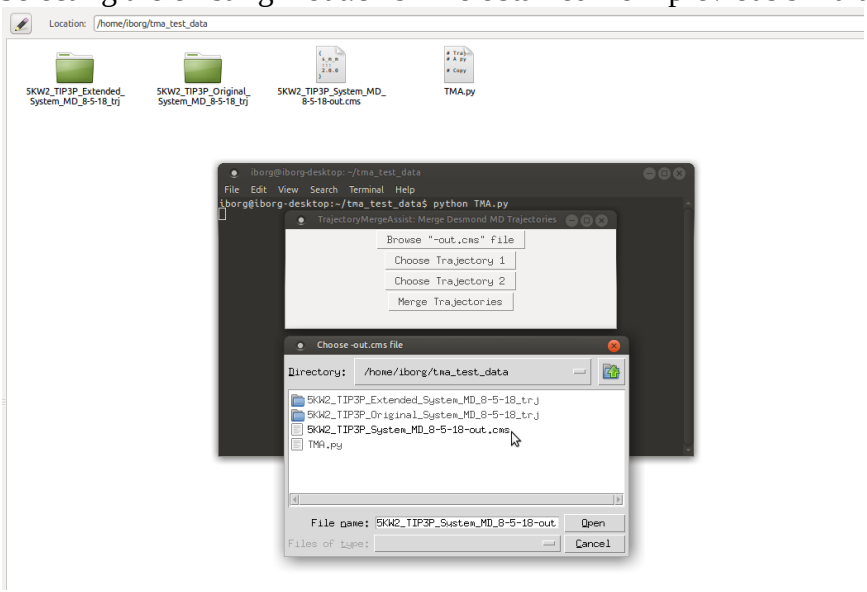
A user-friendly graphical user interface (GUI):

The number of trajectories to choose has been set as two in the GUI, assuming that further extended simulations would be decided based on observing and interpreting the first combined outcome as a statistical plot. Future merging can be done with merged files already made via this tool. It also takes care of merging trajectories irrespective of the version of Desmond installed and also generates an "-out.cms" file instead of ".cms" for version 2018-1 or later which happens during command line usage.

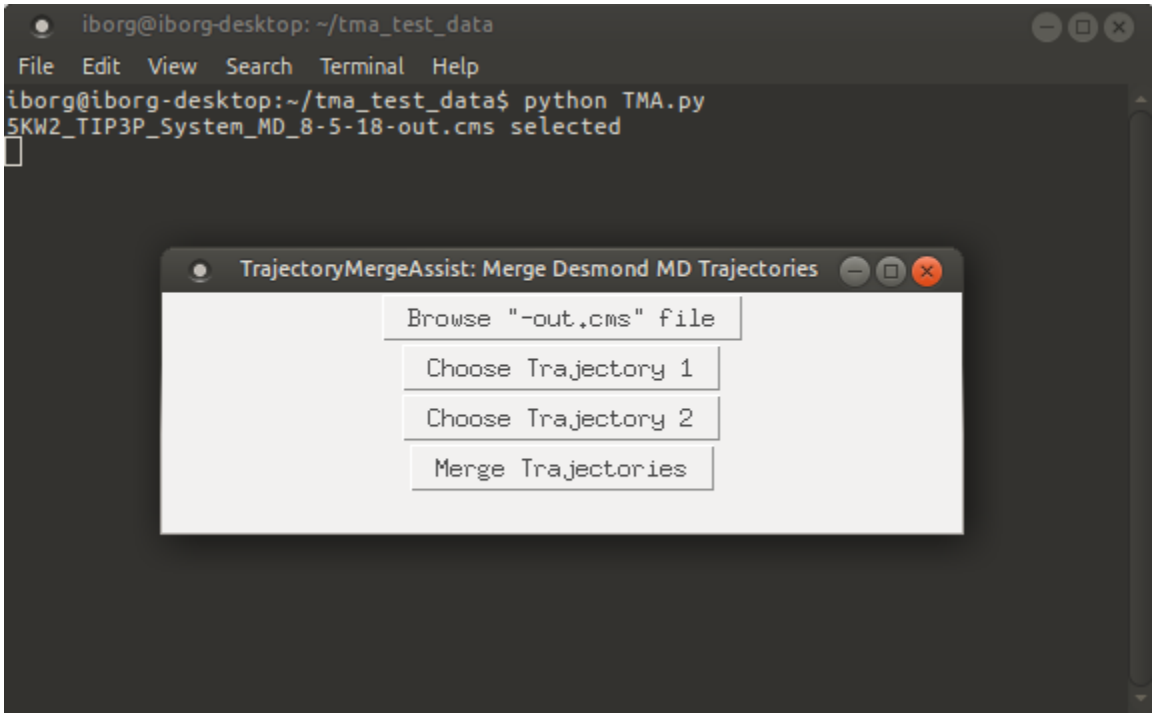


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

Selecting the existing "-out.cms" file obtained from previous simulations:

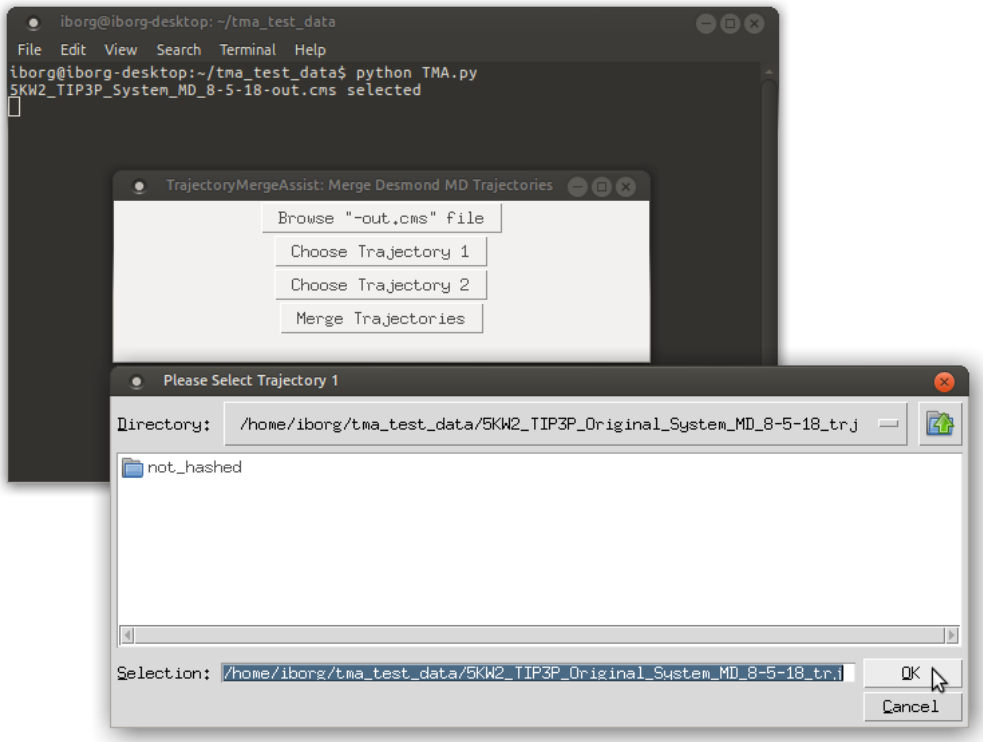


Confirming selection 1:

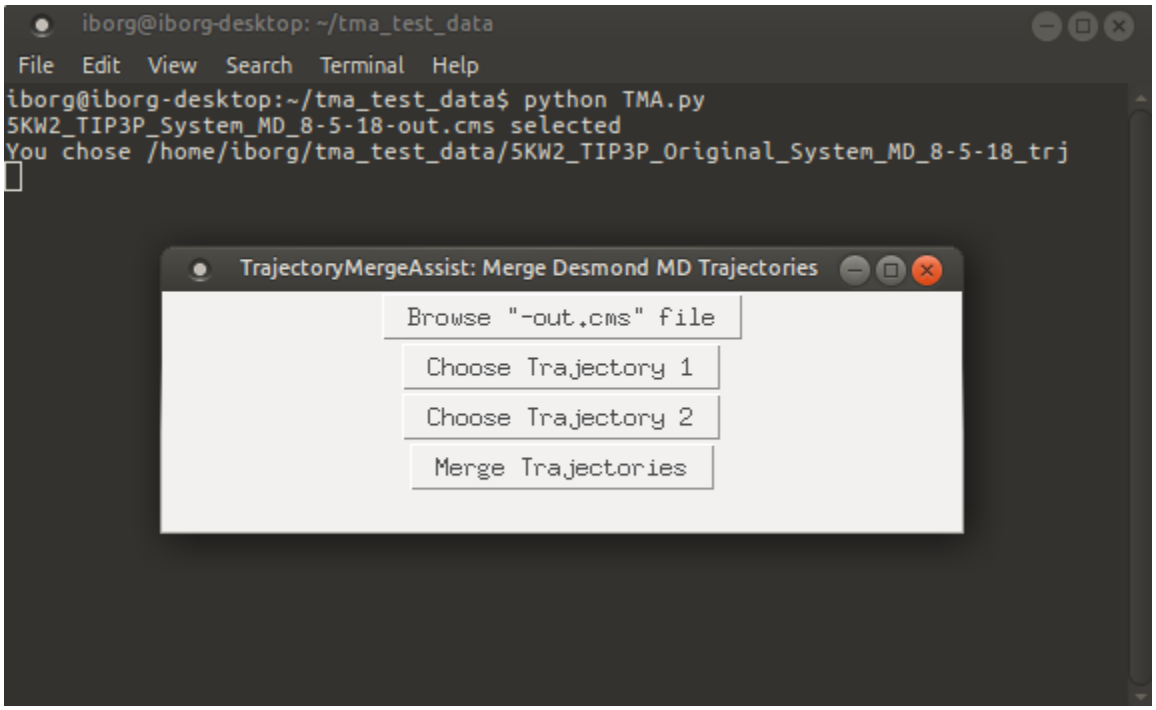


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

94 Choosing first trajectory:



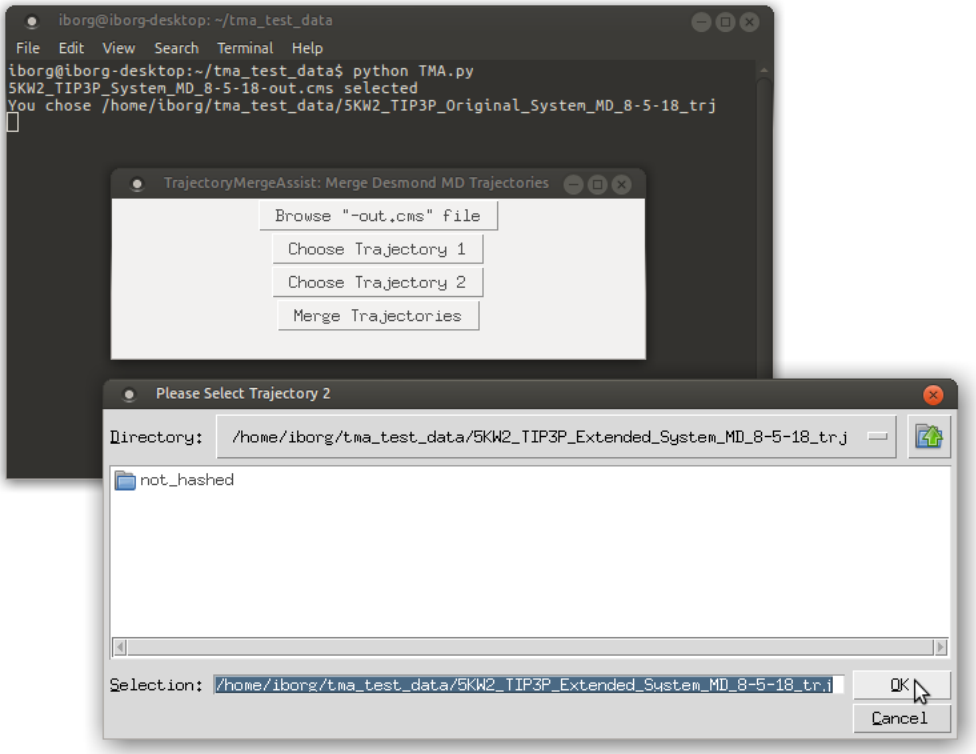
95 Confirmation 2:



96

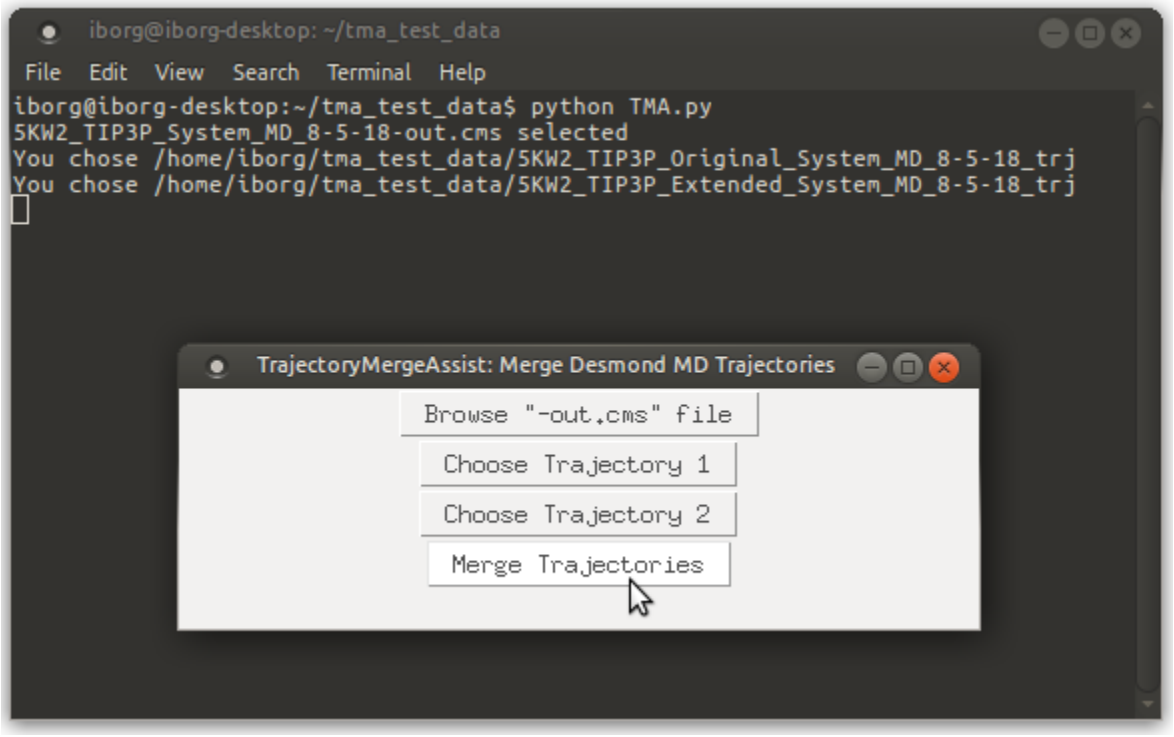
PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

97 Choosing second trajectory:



98

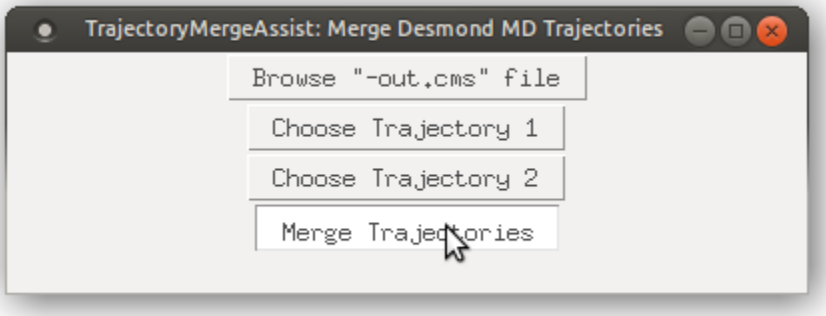
99 Final confirmation for second trajectory selection:



100

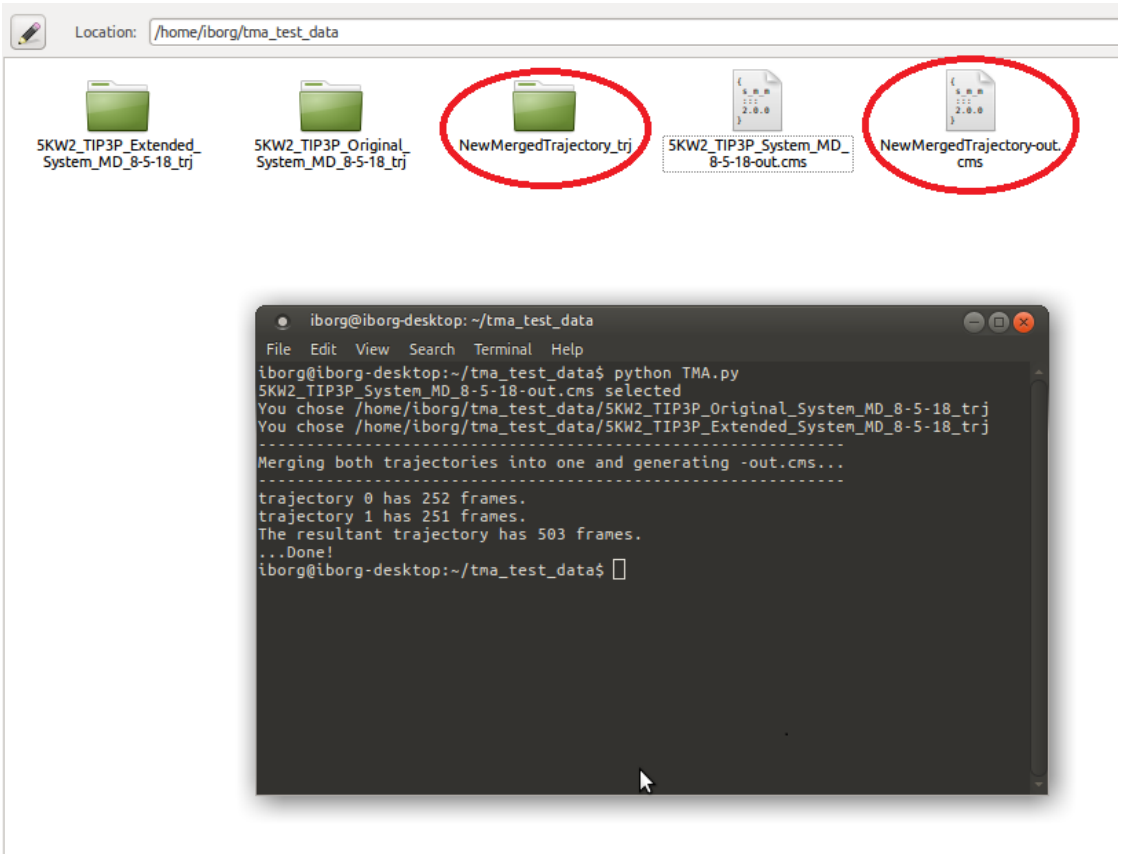
PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

Beginning merge:



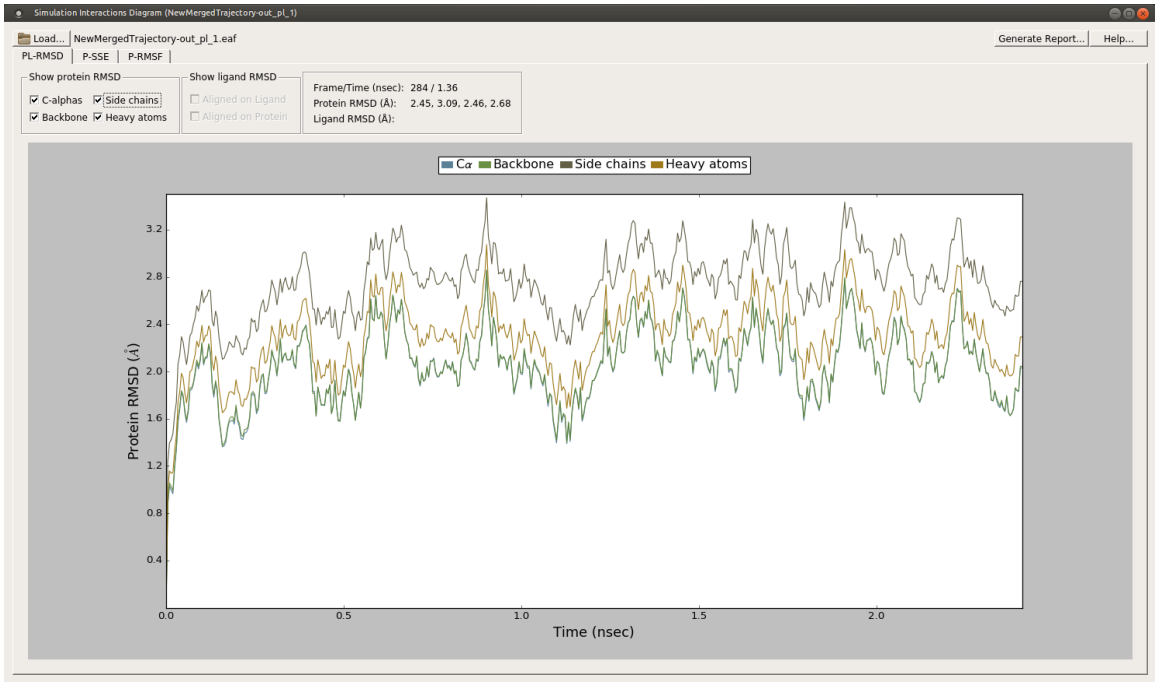
Results

New merged trajectory "_trj" and "-out.cms" file will be created in the same directory. The time taken for completing this procedure depends on the size of the input files (original and extended simulation times):

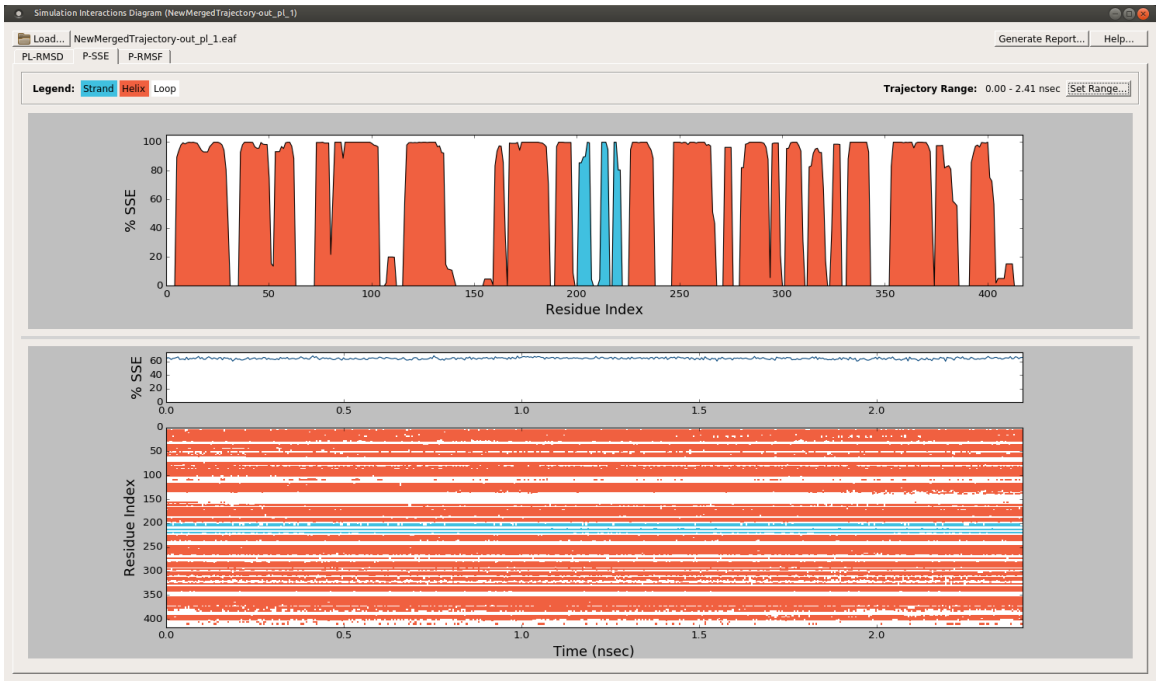


PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

107 Viewing the statistical plots from 0-2.4 ns with the newly obtained trajectory:



108



109

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018

110

111 Conclusions

112 The user need not worry about the corresponding command for merging trajectories as it is taken
113 care of with the help of a simple click on the GUI. This improves usability by a huge extent and
114 makes it very convenient for the user to manage multiple extended trajectories. Additional
115 features such as independent execution regardless of Desmond version and making the output
116 file ready for viewing on Maestro after merging, makes it even more convenient.

117 Acknowledgements

118 • I would like to thank my PhD advisor, Prof. Dipankar Chaudhuri for his moral support and
119 for providing the opportunity to create the tool while conducting molecular dynamics
120 experiments.

121 • I would also like to thank my co-researcher, Argha Mandal for encouraging me for this
122 work.

References

Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, Andrea Cavalli. 2016. "Role of Molecular Dynamics and Related Methods in Drug Discovery" Journal of Medicinal Chemistry 59 (9), 4035-4061. https://doi.org/10.1021/acs.jmedchem.5b01684.

Kevin J. Bowers, Huafeng Xu, Edmond Chow. 2006. "Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters." In Proceedings of the 2006 Acm/Ieee Conference on Supercomputing. SC '06. Tampa, Florida, USA: CiteSeerx. https://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.98.2121.

Robert T. McGibbon, Matthew P. Harrigan, Kyle A. Beauchamp. 2015. "MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories." Biophysical Journal 109 (October). Elsevier:1528–32. https://doi.org/10.1016/j.bpj.2015.08.015.

D. E. Shaw Research, New York, LLC. n.d. "D. E. Shaw Research | Desmond." https://www.deshawresearch.com/resources_desmond.html.

Schrödinger, New York, LLC. n.d. "Schrödinger | Maestro." https://www.schrodinger.com/maestro.

Avimanyu Bandyopadhyay. (2018, May 7). TrajectoryMergeAssist: A Python based utility to simplify extended molecular dynamics simulations on Desmond (Version v1.1.03). Zenodo. http://doi.org/10.5281/zenodo.1243715.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.26920v1 | CC BY 4.0 Open Access | rec: 9 May 2018, publ: 9 May 2018