From: **Mala Radhakrishnan via University of Colorado Boulder Journals** noreply@journals.colorado.edu
Subject: [livecoms] Editor Decision
Date: May 12, 2022 at 4:10 PM
To: Jack Henderson j.a.henderson.1701@gmail.com, Ruibin Liu ruibin.liu@rx.umaryland.edu, Julie Harris juaharri@gmail.com, Yandong Huang ydhuang2727@163.com, Vinicius Martins de Oliveira vmoliveira@rx.umaryland.edu, Jana Shen jana.shen@rx.umaryland.edu

Dear Jack Henderson, Ruibin Liu, Julie Harris, Yandong Huang, Vinicius Martins de Oliveira, Jana Shen:

We have reached a decision regarding your submission to Living Journal of Computational Molecular Science, "A Guide to the Continuous Constant pH Molecular Dynamics Methods in Amber and CHARMM v1.0".

Our decision is: **Revisions Required**
You will see that the reviewers below believe that your tutorial has a lot of strengths and could be a very useful, important, and timely resource for the modeling community, but they bring up some technical issues they had while running parts of it as well as some theoretical framing issues that should be addressed in your revisions.

Please submit a required manuscript to us, ideally within the next month or so (i.e., **by June 12**, though let us know if you need more time, since we know it's a historically busy time for many right now).

The reviews are included below. Thank you, and please let me know if you have any questions,

Mala Radhakrishnan (Lead Tutorials Editor, LiveCoMS)

Reviewer E:
Recommendation: Revisions Required

Significance

> The LiveCoMS Tutorial review " A Guide to the Continuous Constant pH Molecular Dynamics Methods in Amber and CHARMM v[1.0]" by Jana Shen and co-authors is excellent and very detailed. Jana Chen is a recognized expert in the field I am sure that the review will be appreciated by the community.
>
> I have a few remarks and minor revisions, which pertain mainly to the first part (pp. 1 to 5).

Community

> Fine.

Professionalism

> Fine.

Article-Specific Review Criteria

> N.A.

Other Specific Comments for the Author

.

In the introduction, the authors make a distinction between the discrete Monte Carlo algorithm (jumping between protonated and unprotonated states suddenly) and the continuous algorithm like lambda-dynamics that allow any value of the the coupling parameter lambda. This way of presenting the different algorithms is both accurate, but perhaps a few aspects need further emphasis to achieve maximum clarity. It is important to distinguish models (implicit versus explicit solvent) and algorithms (discrete versus continuous lambda).

In my understanding, the discrete algorithm is most clearly realized in implicit solvent models where it is possible to change the protonation states abruptly without encurring large energy changes. Here, both the algorithm (discrete) and the model (implicit) are clear. Implicit solvent is a model that is arguably more empirical and approximate than explicit solvent, but here both the conformational sampling from unbiased dynamics and relative balance of the different protonation states from the constant pH simulation are generated consistently within this implicit solvent model.

The presence of explicit solvent creates a considerable challenge and essentially renders the discrete algorithm essentially impractical. One compromise is to execute the discrete jump of protonation assuming an implicit solvent model, while propagating the unbiased dynamics with explicit solvent. The models here are mixed. The relative balance of protonation states is governed under the implicit solvent, but the conformational sampling results from the explicit solvent. This is certainly an interesting avenue, but the users should be told clearly and unambigusouly that the ionization equilibrium is really that of an implicit solvent model. Please make sure this is clear.

Certainly, lambda-dynamics and the non-equilibrium MD/MC algorithms provide more rigorous solutions to generate constant pH simulation consistently within the explicit solvent model. It is correct to qualify the neMD/MC algorithm as "discrete" in the sense that system only visits the end-states, although is would also be illuminating to point out that the non-equilibrium switches require a continuous coupling parameter lambda like lambda-dynamics. In neMD/MC, lambda is changed from 0 to 1 according to a schedule, while in lambda-dynamics the coupling parameters are free to evolve dynamically. There is some conceptual similarity between lambda-dynamics and neMD/MC and pointing out this similarity may be beneficial to the users.

Page 2

The text says "Although the latter is a major caveat, the CpHMD method offers significantly faster convergence and importantly it can be implemented for implicit-, hybrid-, and fully explicit-solvent simulations." I am not sure why lambda-dynamics can achieve faster convergence than discrete jumps in the case of implicit solvent simulations. Since implicit solvent simulations permit the purest form of the discrete algorithm, what is the advantage to execute lambda-dynamics then? Isn’t a simple discrete jump with MC conceptually simpler than lambda-dynamics (no need to masses, intermediate states, etc)? Based on my understanding of the scripting for CHARMM and AMBER, I would think that if you have the infrastructure to run lambda-MD, converting this to remove the MD on the lambda and replace it by random discrete jumps of lambda between 0 and 1 could be easily realized. Please comment.

Another point about efficiency is that the switch in neMD/MC can be fairly short as concluded by the analysis of Radak and Roux [Radak, B.K. and B. Roux. Efficiency in nonequilibrium molecular dynamics Monte Carlo simulations, J. Chem. Phys. 145,

134109 (2016)]. It is my understanding that what has historically limited the efficiency of the neMD/MC algorithm was the fact that the NAMD program was slow (apparently now resolved).

Page 2

The text says "The all-atom CpHMD method by the Shen group is the only one that includes titratable water to compensate for the net charge fluctuation as a result of proton titration in the CpHMD simulation [48]." In fact, previous implementation of the neMD/MC algorithm also included charge neutrality (Chen and Roux, JCTC 2015) where protonation was coupled to the conversion of a water molecule into a chloride anion, and deprotonation was coupled to the conversion of a potassium into a water molecule (eqs 30 and 31 in their paper). The NAMD implementation of Radak et al (JCTC 2017) supports this feature also, as multiple sites can be protonated and deprotonated simultaneously. However, Chen and Roux showed that charge neutrality acts as a constraint and decreases the acceptance ratio. It is like to have similar effects in lambda-dynamics. Lastly, charge neutrality is not a prerequisite in simulations with PBC and PME (in fact, charge neutrality is necessarily violated during a charging free energy calculation).

Page 2

The text says "While this may well be true when the switch of one protonation state is involved, the problem quickly becomes intractable when multiple titration sites are involved. " Perhaps emphasize that the problem becomes intractable when the number N of sites is large because the number of possible ionization states N goes like 2^N.

Page 3

The equation for the forces on the lambda variable implicitly assumes that there are only changes in charges and Lennard-Jones paramters. But what about other changes in the force field? In the CHARMM force field, dihedral angles often depend on the ionization state of a side chain.

MINOR

In the Abstract: "This is a significant drawback, as" --> perhaps change drawback with limitation?

Page 5

The writing style is clear but perhaps sounds a bit like spoken words with many "We", "you", "our", etc... For example, "Like all computational chemistry calculations, it is important to know your system as much as possible before any serious". Perhaps change by "it is important to know the system of interest" and generally use a slightly less familiar tone.

Reviewer H:
Recommendation: Revisions Required

Significance

> The authors provide a guide for continuous constant pH MD simulations. This is very important as (1) the function and structure of protein systems depend on specific pH; (2) pH is implicitly shown in conventional MD simulations. Overall the guide is very clear and practical. The codes provided are useful and can be applied to general protein systems.

Community

> This work match well with the current MD simulation work. Both CHARMM and AMBER are very popular packages for MD simulations. This constant pH MD simulation method can greatly benefit the community and researchers in related area.

Professionalism

> This manuscript is very well written and easy to understand. It is quite easy to follow the code examples.

Article-Specific Review Criteria

> The authors provide a GitHub link and several url links to specific files on GitHub. The guide is suitable for researchers with some simulation background. The codes can be easily generalized to other proteins.

Other Specific Comments for the Author

> Generally this manuscript is very well written except for some minor revisions.
> 1. There are some inconsistent file names. I noticed two cases: (1) in the lower right code block on page 7, the INP filename should be step2a_xxx; (2) in the first line on page 9, it should be step1b_add_h_crys_xwat.inp.
> 2. There is a weird line in the lower right of page 10 where has a single 'step0.2_equil.inp' hyperlink line.
> 3. There are some broken url links: (1) the url point to GitHub in the first page; (2) the README link in upper left of page 14; (3) the 'cphmd_parm_fit.py.' link in upper right of page 18. I checked the authors LaTeX on GitHub and found out the reason is that the authors have a new line for the web url in \url command. Once I delete that, all url links work well.
> 4. The url links are not consistent. Some file links are actually folder links. It would be good if the authors can reformat the links.

Reviewer J:
Recommendation: Revisions Required

Significance

Significance

There is growing interest in the field in developing simulation techniques that will allow us to better understand the effect of pH as a modulator of several biological process. The tutorial by Henderson et. al. introduces one of such techniques (constant pH MD simulations) with a focus on the Continuous approach, with the objective of generalising a technique that has been successfully used to investigate several systems of biological relevance. They provide instructions to conduct this simulations in two of the most used molecular simulations packages, which is very welcome as setting up these simulations is a tricky process. Therefore I think that this tutorial is a significant contribution that will be of interest of readers of the Live Journal of Computational Molecular Science.

Community

The authors provide a brief introduction to the state of the art of constant pH simulations and position Continum approaches within the wider scope of field. In my opinion the cited references are relevant to the topic at hand and self-citations are only used where relevant.

Professionalism

The manuscript is for the most part well written and easy to follow (with some fragments that are a little bit more obscure to me highlighted in the comments to authors). Figures are clear and highlight relevant aspects of the text.

Article-Specific Review Criteria

Although for the most part I think the tutorial is useful and easy to follow, and complies with the acceptance criteria of the journal, there are some minor aspects that in my opinion should be addressed prior to publication. At this point I would like to highlight that due to constraints arising from other professional obligations and and hardware/software availability, I have only tested the AMBER part of the tutorial and, therefore, the comments below will highlight points that should be addressed on that part, without making assumptions on the technical correctness of the CHARMM part of the tutorial.

Technical errors:
- It seems that the instructions provided by the authors on the gitlab repo about patching AMBER20 are not completely correct.
Specifically, in AMBER20 it seems there is no longer a "src" folder under the $AMBERHOME path that points to the installation folder, and instead all the source files are left in the source folder. Furthermore, attempting to patch the latest distribution of amber20 results in several errors as follow:
"Reversed (or previously applied) patch detected! Assume -R?
X out of X hunks FAILED -- saving rejects to file ./src/cuda/base_simulationConst.cpp.rej"
Pushing throught this errors and attempting to rebuild pmemd resulted in the following cmake error:
"9 errors detected in the compilation of ./src/pmemd/src/cuda/gti_cuda.cu.
CMake Error at pmemd_cuda_DPFP_generated_gtei_cuda.cu.o.RELEASE.cmake:278"
Nevertheless it seems I was able to run the simulations correctly, despite failing in completing the patching of amber20, so I am not completely sure what happened there, but this issue may affect other readers"

- The mdin input files generated by cphmd_prep.sh were actually named with two decimal points, not one as they appear in the tutorial (e.g 3BDC_chainA_A_pH7.50_prod.mdin instead of 3BDC_chainA_A_pH7.5_prod.mdin)

- Attempting to run the mdin files generated by cphmd_prep.sh results in a Fortran runtime error beacuse of an Interger format overflow due to the size of the value of the ntwr flag wich is set to 1*10^10. Reducing the lenght to 1*10^6 gets rid of the error

ntwr flag wich is set to 1.0e-10. Reducing the lenght to 1.0e-8 gets rid of the error, although I did not test the effect on the performance.

Other Specific Comments for the Author

In addition to the technical issues highlighted before, I think that the tutorial will benefit from a more extended explanaition about which keywords to use and modify in AMBER in order to run CpHMD simulations, rather than exclusively focusing on how to use cphmd_prep.sh to set up the simulations. In my opinion, the reliance on that script makes the whole process a bit too much of a blackbox.

It will good if the authors could provide additional context to the sentence "If the objective is to predict the pKa values of soluble proteins or their protonation states at a certain pH condition, GBNeck2-CpHMD [41, 42] is the best choice..." in the first paragraph of the second column in page 5.

In page 6, when discussing set up of CUDA environment, it may be better to use a generic CUDA version as they do with the AMBER version.
