Reviewer(s)' Comments to Author

January 5, 2019

1 Reviewer: 1

We would like to thank the reviewer for their comments as we believe they helped make the manuscript much stronger and a better representation of the work being described. We will address the reviewer's comments in the order they were raised:

> This is timely and a really nice piece of work. This Cis-interface has the potential to become a major tool to support the future development of the plant systems model integration. I have a few specific comments on this and would like the author to address.

☑ First, it would be great to have a more clear description of the novelty of the pipeline developed. Is it mainly a compilation of existing tools, in that case, this needs to be clearly indicated. If not, then a clear statement in this might be great.

While `cis_interface` does make use of several existing tools, it uses them in novel ways and implements several new tools of its own. We have added text to the methods introduction and methods subsections, clarifying the new features that are provided by `cis_interface`. In addition, we have added a new "Advantages & Limitations" subsection to the Summary that explicitly describes the features provided by `cis_interface` that are not available in other existing packages.

☑ Secondly, the work will benefit a lot from a more explicit description of a user case. For example, clearly write out the exact commands needed to finish one round of code integration.

We whole heartedly agree with the reviewer that the manuscript greatly benefits from a concrete use case. As such, we have added a new section "Worked Example" after the methods section that walks through, step-by-step, integration of the example models presented in the introduction. The code from the example is available on the paper branch of the `cis_interface` GitHub repository and will be published as part of the documentation during the 1.0 release.

☑ Thirdly, I would be keen to show some performance data of integration of different modules. No data are provided on this at this stage.

We believe that the reviewer is referring to performance data on integration of specific biological *models*, but would appreciate some clarification if this is not the case. While we would like to show concrete real-world examples of the performance of some integrations done with `cis_interface` using actual models, most of the models that `cis_interface` has been used with are still in the pre-publication stage. As the code for the models is not yet public, these performance tests would not be informative to those without access. In addition, the performance increase due to parallelism and asynchronous communication is highly dependent on how coupled the models are and how many models are part of the integration. If two models are integrated that are tightly coupled, one model will always be waiting for the other to complete its calculation and there not be a speedup over scripted integration of the two models via an intermediary file. However, if one model had other calculations to perform before it needed the data from the other model, a speedup would be possible and completely dependent on the ratio of independent calculations to dependent calculations and the time required for model communication. The speedup calculation is even more complex when more than two models are involved. To reflect this complex relationship, we have added a subsection to the end of the results section entitled "Speedup from Parallelism" that discusses this and present equations for calculating the speedup for an integration of two example models given the appropriate parameters.

☑ Fourthly, the main purpose of this paper is to describe a platform which can help plant modelers of different research areas and using programming languages work together more effectively and easily, it would be great if author can explain the advantage of this new software from the perspective of users.

We have expanded the "CurrentStatus" subsection in the Summary section to more explicitly describe the utility of `cis_interface` to domain scientists, including its ability to promote model reuse, acessiblity, and collaboration.

☑ Fifthly, many of the software engineering methodologies described in the paper are widely used in the field of software engineering. It would be better to provide some more concrete description of how this particular tool help users.

The new "Advantages & Limitations" subsection in the Summary describes the advantages of `cis_interface` over existing software packages targeted at similar problems. This is in addition to updated methods sections that are now more explicit in their description of features which are unique to `cis_interface` and a worked example.

☑ Finally, it would be helpful to provide a flow chart and pseudo-code of the core algorithms used to link different modules.

Although not exhaustive in their representation of the algorithms, we have added pseudo-code for the core algorithms in an Appendix that are intended to provide the reader with a general picture of the processes involved.

2 Reviewer: 2

We would like to thank the reviewer for their comments as we the manuscript is stronger as a result. We have responded to the reviewer's comments in the order they were posed.

> This is a very well written manuscript that is accessible to people from different backgrounds. Few minor comments:

> ☑ Background: Pg 4, L8-18: The first paragraph contains broad, bold statements with no citations as examples. To move away from a largely anecdotal paragraph, examples from the literature should be given. It would also help to cite the Crops in silico perspective article in Frontiers (https://www.frontiersin.org/articles/10.3389/fpls.2017

We agree with the reviewer that the paragraph in question was vague and have added several references to examples from the literature, including the Crops in silico perspective article.

> ☑ P4, L35-38: give examples here too. What solutions?

The sentence in question "Within both computational biology and the larger scientific computing community, groups have developed around different solutions to the problem of connecting models." was the last sentence in the introductory part of the background section and was meant to refer to the solutions described from the literature in the following subsections. We understand the confusion and have clarified this by explicitly listing the solutions that are described in the subsequent subsections.

> ☑ P6 line 17: a to "as"

This typo has been corrected.

> ☑ P7 line 9: remove extra "tools"

This typo has been corrected.

> ☑ In the conclusions, it would be helpful to have a few lines suggesting uses or applications of the Cis interface. For example, researchers could use cis_interface as a user-friendly way to run model code either from the published literature or when reviewing manuscripts. Likewise, experimentalists can update parameter files based on in-house experimental data and then run the models, even if they have no prior expertise in modeling.

We have added several sentences to the conclusions discussing the use cases for `cis_interface` including as a tool for integrating models and as a tool for executing models without knowledge of the model's programming language. While we agree with the reviewer that `cis_interface` could be used for peer review and collaboration by exposing models via a standard interface, we believe this application of `cis_interface` will be most powerful when used with the model integration GUI (which is separate form the `cis_interface` package itself) or by calling models as functions from Jupyter notebooks (which is a feature that was added after submitting the manuscript and will be released in verison 1.0 of `cis_interface`). As such, we have added a subsection for "Models as Functions" in the "Ongoing/Future Improvements" section and discuss this application further there and in the "Graphical User Interface" subsection.

3 Reviewer: 3

We would like to thank the reviewer for their comments as we believe they helped make the manuscript much stronger and a better representation of the work being described. We will address the reviewer's comments in the order they were raised:

> This manuscript reports on efforts to create a universal interface toolbox for integrating computational models across languages and scales. This topic is without doubt one of the key topics in further development and application of existing and to-be-developed plant models. I appreciate the authors? effort in developing an open source Python package cis_interface, which can be installed on 3 different operating systems and support integration of models written in 4 different programming languages. I think the paper has great potential but I do have some comments:

☑

> The author had a lengthy discussion on comparison of communication time cost for model integration with different communication mechanisms, programming languages, Python versions and operating systems, actually this is also the main result of current manuscript. However, without comparison with existed tools, I think this part should be greatly condensed. I believe readers will be more interested in how does cis_interface work for combining different types of biological models into a complex network. These models may have different time step length (e.g. ODE vs constant step length) and different types and scales of input and output which need complex data conversion (e.g. total leaf nitrogen content to different protein levels). Moreover, how to perform integration of input for one model from different models? For example, canopy model only gives 3D mesh data with light level on each facet, however, metabolic model needs both light level and protein levels (calculated from another model) for each facet, how to correctly integrate the protein data and the light data?

We thank the reviewer for this comment as it has pointed out several areas in which the manuscript was previously lacking and major improvements have been made as a result. First, we have added a subsection entitled “Speedup from Parallelism” to the end of the “Results” section that discusses the speedup achievable using `cis_interface` for a two model integration based on model properties and the presented performance results and we have added a subsection entitled “Advantages & Limitations” to the “Summary” which enumerates the advantages of using `cis_interface` over other existing tools targeted at similar problems.

Second, we have added a section “Worked Example” which walks through each step required for an integration of the two example models presented in the introduction. All of the code and YAML files required for this integration are included in the text, are available on the `paper` branch of the `cis_interface` GitHub repository, and will be released with version 1.0 of the code. While we agree that a more complex worked example would be more useful to users who are already familiar with the code, we believe that such an example is outside the scope of this paper which serves as a introduction to the goals, methods, and performance of `cis_interface`. There are several more complex examples provided online in the `cis_interface` documentation (including an example from the 2018 Crops in Silico hackathon using toy biological models), which we now reference throughout the paper such that readers can refer to them.

Third, many of the topics that the reviewer explicitly mentions (time steps, complex data conversion, data aggregation from multiple sources) are problems that are not completely solved in the current version of `cis_interface`, but are solved or ameliorated by ongoing work (discussed in the “Ongoing/Future Improvements” subsection of the “Summary” section) that will be released in version 1.0. To address these concerns we have dedicated the last subsection of the “Worked Example” section to “Advanced Integration Topics” which discusses these use cases, references examples in the documentation, and points to the appropriate topics in the “Ongoing/Future Improvements” subsection. In addition, we have expanded the “Ongoing/Future Improvements” subsection to provide additional details and also identify those improvements which will be released in version 1.0. Those features not yet supported by `cis_interface` are also discussed in the new “Advantages & Limitations” subsection.

☑ I expect more detailed biological modeling examples to be provided, from which the readers are able to know how to use the cis_interface and what’s the advantage of it. I also expect the author to suggest some general modeling principles from following which the newly developed model can easily be integrated with other models.

As discussed in our response to the reviewers previous comment, we have added a new “Worked Example” section which provides a detailed walk through of integrating two example models. Throughout this section we provide tips on creating/preparing models such that they can be easily integrated using `cis_interface` and follow best practices for reproducibility. Although we would also like to include real examples from the integration network shown in the "Summary" section, most of the models in that network are in the pre-publication stage and their source code cannot be released at this time. Although not all biological models, we do reference several more complex examples from the `cis_interface` documentation throughout the text. In addition, we have expanded the summary to include a more in-depth discussion of use cases and the "Advantages & Limitations" subsection discussing the advantages of `cis_interface` over other solutions.

☑ Page 7 line 9: remove redundant 'tools'

This typo has been corrected.