

Response to Review

Associate Editor

The main issue regards the motivation (reviewer 1) and target audience (reviewer 2) of the application. If this is meant for researchers that are proficient at modeling and coding, it should be made clearer how they benefit from this automated application (wouldn't they already have similar protocols and pipelines in place in their routine research?). If instead, this is for empiricists, or environmental managers, with little coding experience, the manuscript is too technical and the application perhaps should be made more 'user-friendly' (reviewer 1). My suspicion (and reviewer 2's) is that it aims for that increasingly common middle person (the computationally literate ecologist). However, as Reviewer 2 points out, it is not clear how such audience should interact with the defined pipeline and how, for example, they could propose new models. A fourth (very attractive) possibility is that this is meant as an interface for people with different skills and motivations to collaborate (e.g. to foster the feedback between scientific development and applied use). If this is the case, the protocols of interaction need to be clarified, as well as the audience for this article (e.g. is it the computational ecologist who will read this article and propose applied scientists to work with her/him through this system?). Defining the audience clearly should guide how to improve the application and manuscript in a way that it ensures its use and long-term survival as an important short-term forecasting tool.

We have expanded the last paragraph of the Introduction to address the question of target audience and split the opening paragraph of the Introduction into two paragraphs to provide expanded treatment of the motivation for this approach including the addition of three examples.

The AE and reviewer are correct that we have target this paper primarily at the "increasingly common middle person" and "researchers that are proficient at modeling and coding". Thanks to the expanded efforts at computational training we believe that many researchers have the necessary skills to implement these sorts of approaches and we think that the association between "empiricists" and "little coding experience" is increasingly out of date as demonstrated by the fact that this system was developed in large part by the empiricists collecting the data.

That said, we certainly recognize that not all ecologists and evolutionary biologists have the necessary skills to implement this sort of pipeline (just as many do not have the skills to implement a Bayesian statistical analysis). As a result, a single paper in this space won't be perfect for everyone. We have tried hard to reach a middle ground where users with the necessary skills to implement this sort of system will be able to understand precisely what we did while also providing introduction and motivation for researchers with less advanced computational skills to learn more and collaborate. To reach researchers with relatively basic computational skills we have recently completed a paper focused on the data management component of this pipeline that focuses on providing a ground up introduction to this kind of automation in an ecological context (<https://doi.org/10.1101/344804>). We hope that the combination of these two papers will successfully introduce this general class of approaches and concepts to a broad array of researchers.

We disagree with the assumption that researchers that are proficient at modeling and coding would already have similar protocols and pipelines in place in their routine research?). We are active in this field and aware of very few efforts of this scope in ecology. These researchers will have the foundational tools to develop similar systems (or contribute to this one).

Reviewer: 1

Overall this is a clear (and very thorough) description of a useful set of methods and tools for addressing a compelling need in ecology. In this age of increased skepticism of science among the general public and increased awareness among scientists of the mischief that statistics can cause, this kind of automatically updating system seems like the most honest and open approach to science possible.

Thanks!

My only real comment/question regards the motivation for the automatic and iterative nature of this. The full data processing to forecasting pipeline is great because it integrates data processing, model fitting and model comparison. But I suppose the skeptical reader might wonder, can't all of this be done without automatic and iterative part? Can't we compare model predictions the old fashion way, using held out data, not by forecasting truly novel data? The response might be that forecasts themselves are inherently useful for planning. Again the skeptic might question whether there really is a demand for such short-term forecasts? Maybe more examples could be given. My hunch is that the real benefit of iteratively adding truly novel data is that it makes the science as honest as possible. I guess I'm asking that the text somewhere address the skeptic's questions above.

There is definitely a need and demand for short-term forecasting. This was laid out in detail in Dietze et al. (2018; PNAS), but we agree that the current paper benefits from providing more direct treatment of this. We have significantly expanded the opening paragraphs of the introduction to provide specific examples and further emphasize the value of this approach.

I might also suggest differentiating parameter estimation and model development. Forecasts can improve by adding new models, or by adding new data which then improves parameter estimation in the old models. The automatic pipeline is great at the second part of this, continually adding data and re-fitting the old models to the new data and then producing new forecasts. However, since there was so much data in the Portal project to begin with, I doubt that parameter estimates in the models have changed much since the automatic pipeline was established (maybe they have?). However, in a new project that was just getting started (e.g. NEON), the main improvements in forecast skill early on would be in adding more data. Eventually, however, the models might stop improving and reach some prediction limit imposed by the structure of the model. As the authors point out in the discussion, finding this point will be key for improving ecological theory. I think a lot of ecologists think their models are useful, they just need better parameter estimates. My sense is that actually, most ecological models are just crap for forecasting, it's not for lack of data! Maybe this is an obvious point, but the role of improved parameter estimation in this whole endeavor kind of gets lost for me. Does the pipeline even save the fitted model parameters? Would that be worthwhile?

There are a number of interesting points in this paragraph that we feel are beyond the scope of this current paper. We have already described the importance of new models and how we have facilitated adding them. We have now further expanded this section to describe how those external to the group can contribute improved models. Introducing and discussing the different implications of different forms of error and how they are associated with increasing in data vs. improvements in models is an interesting subject for forecasting, but is out of scope of this paper. It would definitely be something interesting to explore in future work.

In Figure 1, I feel that something is missing here about how other scientists are supposed to interact with the data and forecasts. The manuscript makes it pretty clear that this is not just for internal use, but is it actually inviting other labs to add models and make predictions? How specifically would another lab contribute a model to the project? Would they fork the project on github, develop a new model and then do a pull request? Or would they create a separate prediction archive themselves? Maybe add an arrow in Fig 1. showing how new

models can be plugged into the forecasting work flow. Box 1 addresses some of this but I guess I want more details.

We have added description of how others can contribute to the Modeling and Forecasting section of the paper. We tried to also add this to the figure, but could not find a way to do so without making the figure overly crowded and complicated.

Some other miscellaneous questions: Can models be dropped from the ensemble?

This is not currently done at the moment, but is on the development roadmap.

What if old data needs to be corrected or updated? That could change everything, including the interpretation of the archived predictions. Is there a way to handle that?

This is a great question about a tricky problem in forecast archiving that we have encountered and are still trying to determine out the optimal way to address. One can update the previous forecast files and use the versioning to maintain a copy of the forecasts that contained the underlying error or leave them as they are and store supplementary data indicating that forecasts prior to the selection should not be used in later analyses. Since we aren't happy with any of the current solutions, and this is a fairly detailed point related to archiving, we think it is best to leave this for later discussion.

Minor comments:

Lines 63 to 83: Could this section include some examples that demonstrate the value of iterative near-term forecasting from other fields? Could it be argued that iterative near term forecasting has led to progress in other fields? Say meteorology and political science?

We have now mentioned the established benefits for iterative near term forecasting in meteorology and added several citations for readers interested in exploring this further.

Figure 1: Another point on figure one. What about model improvement? It seems like there should be a feedback loop somewhere showing how models get refined, added, dropped etc.

We tried to add a visual representation of this to Figure 1, but could not make it work without making the figure overly crowded. We have added some additional discussion of this "Modeling and Forecasting" section of the paper.

There's a lot of technical terms and software names. Worth adding a glossary or table listing all the tools and acronyms used in the paper. Github, Travis, CI, QA/QC, Docker, Rocker, Software containers, portalData, portalr, portalPredictions, Zenodo, data latency, etc.

We have added a glossary.

Figure labels are doubled.

Fixed.

Travis and github icons are very small in Fig. 1

We have increased the size of the icons.

****comment on the website: portal.naturecast.org/evaluation.html – looks like there's a word missing in the caption explaining the coverage graphs. "Coverage: ... equal to ."** ***

Fixed. Thanks.

comment on the website: <http://portal.naturecast.org/index.html> – Figures need more information/captions. Which model is producing the blue lines (the forecasts). Is this the ensemble model?

Yes. We have clarified this on the website.

Line 163: This section here about dependencies seems a little out of place, or at least the transition from the CI section to the details of dependencies management seems abrupt.

There are two main components to Continuous Analysis as defined by Beaulieu-Jones & Greene (2017). We have one paragraph on each in this section on Continuous Analysis. We're not sure how else to handle this, but if the reviewer has any specific suggestions we're happy to consider them.

Figure 2: The figure is pixelated and hard to read (might just be the draft version I have).

The raw images are all high resolution so presumably this will be fixed in production if the paper is accepted.

Reviewer: 2

I enjoyed reading the manuscript by White et al, which provides a very clear description of using cutting-edge software development technologies for ecology by automating near-term forecasting. The manuscript is well thought-out, and well-written and I think should be accepted for publication. I have a few minor line-by-line suggestions and a broader question/theme that was in my mind throughout reading the article that I would suggest the authors consider.

Thanks!

Minor line-by-line suggestions:

L48: Could you cite a few here, maybe give a sentence or two overview of what they do/what questions they seek to answer (are we talking about continuous temperature sensors, camera traps, terrestrial/aquatic/marine?)

We have split the first paragraph into two to allow us to add multiple examples and associated citations.

L102: Like L48 above, please provide citations and examples with this statement

We have added examples and citations to this sentence.

Fig 1:

- **The first time the reader sees "GitHub" and "Travis":** please add citations and descriptions (and/or reshuffle when Fig 1 is cited within the text so you can describe it first in the text)

There is a challenging order issue here. We want readers to be able to see the stages of the forecasting pipeline before delving into the details. This is Figure 1a, which is referenced in the text before the definitions of Travis & GitHub. That section of the figure/legend does not reference those terms. Travis was cited and described before Figure 1b was referenced. We have updated the text so that this is also true for GitHub. We feel that this is the best compromise in that it keeps the two directly related pieces of Figure 1 together, allows the reader to see the big picture of the pipeline before diving into the details, and introduces the associated terms before the relevant section of the figure is referenced in the text.

- **Consistency:** the legend describes the Travis icon but not the GitHub icon

Added.

- **A, B are not labeled in the figure**

Fixed.

This paper is written to quite a narrow audience: ecologists that already code and could benefit from this kind of automation. This is fine. But many researchers do not yet have the coding skills to benefit from automation (the audiences for Software Carpentry, who the authors mention). To me, it seems that in ecology there are two computational camps at the moment: ecologists who are ready for the workflows mentioned in this paper, and those beginning to engage through Software Carpentry. This is obviously over-simplified, but as I read this manuscript I found myself thinking that it would be nice to try to speak to both camps, instead of just the one with the skills for this automation. How do people get there? How do we get ecologists to the point where they too can develop a workflow like the one in this manuscript?

A single paper in this space won't be perfect for everyone. We have tried hard to reach a middle ground where users with the necessary skills to implement this sort of system will be able to understand precisely what we did while also providing introduction and motivation for researchers with less advanced computational skills to learn more and collaborate. Thankfully given recently expanded efforts to provide more advanced computational training to scientists we disagree that it is currently written for "quite a narrow audience". That said, we agree with the reviewer that reaching out to researchers with relatively basic computational skills is important. To address this broader need for introductory material on automation and continuous integration in ecology, we have recently completed a paper focused on the management of data that is regularly updated, which focuses on communicating at an introductory level, including the inclusion of a tutorial for helping beginners to this area get started (<https://doi.org/10.1101/344804>). We hope that the combination of these two papers will successfully introduce these approaches and concepts to a broad array of researchers.

I was happy to see the discussion of team science (~L421), but training and skill-building immediately came to mind, and I didn't know who the "team" was, it seems that it is the Weecology team, not including the Portal Project. Can the Portal Project maintain this on their own? I still didn't have a sense whether Florida researchers created something for Arizona field scientists that will be maintained by the Florida scientists, or whether the Arizona group was trained to maintain it on their own. My assumption is that the Arizona team would be in the Software Carpentry camp, and I'm thinking about how to bridge these camps and empower more scientists with the skills to be able to understand the integrative process the manuscript describes. The goal is to get all ecologists into the automation camp, how do we do it?

I would suggest adding some sentences that address this to the intro and discussion. This could traditionally be seen as out of the scope of the manuscript, but that will continue to perpetuate the two camps instead of bringing the Software Carpentry camp towards the automation camp.

As is already described in the paper, the field researchers are an active part of the team that develops and runs this system. Both components of the team are part of Weecology. As we've mentioned above the clear distinction between empiricists and computational folks really isn't valid anymore and the field researchers involved in this project all have strong computational skills. We have modified the language in this section to clarify how the team works. While we agree that "The goal is to get all ecologists into the automation camp", and we are working hard on multiple fronts to make this a reality, this is a larger challenge that is beyond the scope of this current paper.