Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-76-RC1, 2020

Geoscientific Model Development Discussions

Interactive comment on "TraceME (v1.0) – An online Traceability analysis system for Model Evaluation on land carbon dynamics" *by* Jian Zhou et al.

Anonymous Referee #1

Received and published: 15 June 2020

General Comments

This paper discusses the so-called traceability analysis that the authors have published and applied since 2013. The motivation for the paper is the development of a cloud-based software built on the CAFE framework which implements their method and is launched via web interface. The authors position their analysis as the answer to what they consider are the deficiencies in the current state of model evaluation, describe how their software and algorithms work, and then show analysis results from a selection of CMIP6 models.

I have two reservations against this paper. The first is that the novel portion of the pa-

per, the TraceME software, is not a substantial advance in modeling science. TraceME is a web interface to an analysis script which runs on their server. There is some sophistication in that source data can exist on multiple nodes, but this is the CAFE framework and not TraceME itself. The authors neither provide a link to test out the operability of the software they describe or even a screenshot of the user interface. The reader is left to trust the authors that this software exists and is functional as they say it is.

My second reservation is that a significant portion of the paper puts forth a viewpoint of the state and needs of model evaluation that is poorly supported. The authors use ambiguous terms such as 'traceable', 'shareable', 'indirect effects','high computational cost', and 'automatic'. They use these terms to describe their viewpoint on the deficiencies of current model evaluation, but do not say in detail what these terms mean.

In what follows I will specifically refer to lines which have led me to this opinion of their work.

Specific Comments

lines 22-24: You assert that the main challenge of using observations to evaluate ESMs is the untraceability of model outputs. It is not clear to me what this means precisely or why it is true. Why is it the 'main' challenge among many others?

lines 44-45: What does 'generally treat all metrics equally' mean? The referenced ILAMB package, for example, does not treat every statistical measure equally. Neither does it treat measures from all data sources equally. Also, what do you mean by 'indirect effects'? ILAMB also considers variable-to-variable relationships including metrics such as Koven's inferred carbon turnover time [1]. I am having trouble envisioning what the authors mean by this statement.

lines 56-59: It does not follow that 'an automated computation and shareable platform' is essential because of a increase in the amount of data. Environmental computing

has utilized computing centers for many decades in response to computational and data costs. Furthermore, these computing centers are becoming more user-friendly. For example, NERSC now supports Jupyter notebooks [2] which allow you to script analysis on your browser without needing to move data around. There are also cloud resources which give compute and storage capacity to anyone at low or no cost. This is a trend across many disciplines and even in the private sector.

lines 99-102: This point is misleading. You also are downloading large volumes of data, you are just automating it for the user. They would still need to wait while it downloads or they would benefit from you having pre-downloaded it for them. This is how the community does analysis already. Users can download data once into a project group directory on an institutional cluster where many scientist can perform their analysis. In the case of the CMIP6 archive, much of it has been copied onto NERSC drives where it is directly available to the community via a Jupyter notebook interface. There was even a multi-institutional hackathon [3] to collectively work to push results out faster. The point is that there are many ways around the need of downloading large amounts of data. If access to these institutional clusters is an issue, this is a need that the community should address.

line 118: Jupyter notebooks [4] are another widely used solution which you should reference.

lines 141ff: It is interesting that CAFE can deal with data sitting in different locations. However, I wonder how scalable this idea is. If the required data is large, then the runtime of TraceME will be dominated by download times. This may be acceptable for a relatively small analysis (few variables for a few models at monthly resolution), but could be on the order of days/months if higher temporal frequency is to be analyzed.

lines 152ff: Where is the web-based UI? It feels strange to see you advertise this 'shareable' technology and then not have access to explore just what it is. What are the limitations of what I can trace? Does it depend on what you have previously down-

loaded? Can I upload my own model output? Can I edit the analysis script that is run? Or does it rather run on the limited models you have predownloaded and only the analysis you have setup? If this is the case, a web UI seems superfluous. You could simply upload all possible results to a website for community perusal. In fact, this is what ILAMB does and how the service is most used.

lines 234-235: So the data must be moved to the central node, doesn't this mean download times will dominate your analysis? How is this computationally efficient?

lines 244-245: Does this mean that we are restricted to using models as they were uploaded a year ago? A lot of model data has been uploaded and updated. Or will the web execution of TraceME automatically query a search of ESGF and redownload these model outputs?

line 360: You should expand on what you mean by each of these terms.

* What does 'traceable' mean beyond the execution of your analysis? Why is this aspect of model evaluation so critical? * Does 'automatic' mean executable from a web form? If so, ILAMB has had this for 2 years in the work done by Mark Piper [5]. Also, on each commit to the master branch, ILAMB deploys automatically on Azure-pipelines [6], downloads observational data, and runs a test. Also as you mention, both ILAMB and ESMValTool have workflow that make the (parallel) computation of a huge suite of model evaluations automatic. If this is not what you consider 'automatic', then what is 'automatic' and why is what the community is doing insufficient? * What does 'shareable' mean? The ILAMB package generates a hierarchy of evaluation results that are browseable in a web page that you can distribute to the world by simply uploading it to a web-accessible location. If that is not 'shareable', what is and why is it so important to model evaluation? Furthermore, not every group wants a shareable solution, say for quick verification tests they do not want accessible.

lines 385f: I disagree with you that model evaluation needs to be more efficient. ILAMB may take a long time in serial execution, but this is why it was written to launch in

parallel on several institutional clusters or even a laptop/workstation. I am aware that the entire ILAMB CMIP5v6 comparison runs in a few hours. Given the decadal span between MIPs, I contend that the speed of our analysis is not the bottleneck. Beyond this, there are scripting tools and packages specifically designed to handle parallel and fast evaluation (see dask [7] and xarray [8] among others).

line 387: You argue that there is an 'absence' of automation and then explain how ILAMB and ESMValTool both implement it?

line 396: Unfortunately there is no substitution for technical training. You can setup a system like TraceME which automatically runs analysis. Yet someone has to setup and maintain that system. As software stacks change, it will break. Models will need to be added and updated. The analysis script will need to change. Others will want to upload their own scripts. How will they do this? There is a great amount of technical work that is needed to keep such a setup running and useful. What you have done is made running a relatively narrow task simple, which is by far the easiest part of the work.

lines 404ff: You have not solved the issue of data transfer, you have hidden it. And it is not really hidden either. When the user clicks on your web interface and then has to wait, perhaps days, while the data is downloaded to your central node, it will not feel terribly automatic.

[1] Koven, Hugelius, Lawrence, Wieder, Higher climatological temperature sensitivity of soil carbon in cold than warm climates. Nature Climate Change, October 2017, doi: 10.1038/NCLIMATE3421 [2] https://docs.nersc.gov/connect/jupyter/ [3] https://eos.org/science-updates/hackathon-speeds-progress-toward-climate-model-collaboration [4] https://jupyter.org/ [5] https://permamodel.github.io/pbs/ [6] https://azure.microsoft.com/en-us/services/devops/pipelines/ [7] https://dask.org/ [8] http://xarray.pydata.org/en/stable/

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-76,

2020.