December 15, 2020 Joint Methods & CI Working Group Call

Attendees: Mike Dietze, Quinn Thomas, Bruce Wilson, Carl Boettiger, Alexey Shiklomanov, Jody Peters

Agenda/Notes:

1. Fill out the **Poll** to find a time for recurring Jan-May calls - ignore the dates and just focus on the days and times that work for you.

2. Updates on EFI Task Views
    a. Use Task View 1 on Reproducible Workflows as a guide
        i. How much effort would it take to convert the reproducibility task view to a special issue from the EFI webpage?
        ii. **MDPI Forecasting Journal Special Issue** deadline is March 2022
        iii. Alexey thinks it could be doable in 14 months
        iv. Would this overview/review piece be something MDPI Forecasting is open to?
            1. Good question. Need to look at what the journal accepts. It fits within the scope of the special issue. Expect MDPI would allow it to go out for review.
        v. Do we want it to be the hypothetical that we have now? Or do we want to include case studies that are part of the Forecast Challenge?
        vi. Question for Carl on publishing the server ideas. This is higher priority than the Task View. It is more on theme than the Task View.
        vii. Server is not going to be novel. It is standard infrastructure. Not inventing something new. If it is within scope then sure, but don't want to sink a lot of time in writing if it doesn't fit the scope.
        viii. Could focus it on advertising that there is a reproducible stack that can be used for other things and illustrate how it is used in the Challenge.
        ix. Think about what next things that are practical to get done by the end of the summer
            1. Don't have any tooling around. Tried to enable it so you can go back in time to see what forecast changed because the data changed.
            2. This isn't easy now, it is a digging exercises, but could make
            3. Snapshotting, publishing out to DataOne or other data repositories - still hypothetical.
            4. When revisiting in a year and have forecasts than want to get published then could fleshed this out more.
            5. What Carl thinks is straightforward and standard may not be on others radars.
                a. How things have been chosen to be standardized, choosing the buckets, etc

b. The pieces coming together to build a server is straightforward, but the choices to set up a forecasting challenge is not as straightforward - there are decision points.

c. The “not fragile” idea is important. Central to STC pitch is that there are lot of teams that are building their own workflows that are fragile. If there was something people could grab and run with to accelerate the automation of their forecasts - this would be useful for the eco forecasting community and extended beyond the community to other communities

d. More of an engineering mindset. Putting together pieces/bricks that already exist

e. Advantage of this first special issue in the forecasting issue vs to the ESA special issue. To ESA issue it would be oversell. For forecasting community would be an easier sell.

f. Why not to build custom, one-off solutions to every problem

g. Quinn, Carl to lead on this for the March 2022 deadline

h. Idea for this Issue as well

  i. Challenge Overview paper

  ii. CI - (or if it doesn’t stand up on its own, could be part of the overview - but bet it can stand alone)

  iii. Introduction to Themes

6. ESA Bulletin could be a place for the Task Views as could the Computational Biology 10 Simple Rules

7. Want to get the Special Issue on the radar on all the working groups

x. Bruce shared this site for Common Workflow Language which in a nutshell is:

  1. An open standard for describing analysis workflows and tools in a way that makes them portable and scalable across a variety of software and hardware environments, from workstations to cluster, cloud, and high performance computing (HPC) environments. CWL is designed to meet the needs of data-intensive science, such as Bioinformatics, Medical Imaging, Astronomy, High Energy Physics, and Machine Learning.

b. Uncertainty Quantification & Propagation, Modeling & Stats

  i. Updates – save for the next call

c. Visualization/Decision Support Tools, User Interface

  i. Placeholder while Whitney takes her prelims

d. Data Ingest, Cleaning, Management

        i. Placeholder until we are further along with the other Task Views or have an identified leader for this

3. Forecast Data Visualization Shiny App Update
    a. Bruce Wilson and the ORNL DAAC's experience moving from RShiny to Python Flask
        i. Two key factors: RShiny didn't scale for us (and we were seeing what we think is a memory leak). Airborne Data Visualizer (https://actamerica.ornl.gov/visualize/) was the only thing in our production architecture using it. We wrote this for one particular campaign. Needed to make this work for two more Airborne data campaigns.
        ii. Refactored into Python Flask to make it more consistent with where we are headed for overall systems architecture. We also understand better how to containerize Flask and move it to AWS Kubernetes equivalent.
        iii. Others may have better experience, but we lacked technical depth in RShiny. Definitely a useful prototyping tool
        iv. Made the decision to focus on Python as platform. Ran into issues probably memory leaks. Needing to reboot the server each week that was an issue.
        v. System load was lower on Flask compared to Shiny - easier to replicate visualizer as a matter of configuration.
        vi. It is the same concept as Shiny - just done in Python. Users wouldn't know the difference. It looks essentially the same. They are calling the same libraries.
        vii. Is there a Python Flask server that makes it easy to deploy in the cloud? DAAC is doing it.
        viii. The driving force is R competence and number of things people are competent is. For EFI worried about the other way. Have more people R competent than Python competent.
        ix. For the communities that Bruce serve there are some that are dominantly using R vs those dominantly using Python.
        x. Shiny and Flask are about reactive programming - in principle could set up a minimal Flask instance that calls R script to create visualizations. Infrastructure is Flask based, but code is R. There should be a hacky way to support both. But that isn't that create.
        xi. We are talking about solving 2 different problems.
            1. EFI needs to meet scientists where they are. DAAC needs to manage a web interface.
        xii. At this stage we are dealing with something community generated and maintained by a community based on R. So think we need to take the hit with performance because that is what the community uses
            1. Want to make sure our community isn't niching ourselves into a corner because of a language bias

4. Update from Quinn on NIMBLE
    a. Quinn has been poking around for NIMBLE for class and for the Challenge
    b. Monte Carlo simulations. You can't restart Kalman filter where it left off.
    c. Quinn posted on the user group and they responded within 6 hours with a fix.
    d. Could be a lot of potential to think about where tools are falling short for ecological forecasting applications
    e. Think about where there could be collaborative efforts.
    f. Quinn deployed random walk null model in JAGS and NIMBLE - got two different results. They implement in different ways. Can't use NIMBLE for persistence nulls.
    g. Could schedule chat with NIMBLE
    h. Is there a framework we can fold in a community to see if it has what we want
    i. NIMBLE can handle giving it compiled code as the process model. Can't be stand alone executable. But can write a more complicated model. Couldn't call CLM or an executable you have already compiled
    j. NIMBLE is not ready for ecological forecasting applications yet.
        i. It is and it isn't.
        ii. Backbone for Mike's flux forecast. Analysis step is benign done in NIMBLE. Could make that more efficient and more usable in a general sense.
        iii. NIMBLE is a natural choice because it is more adaptable than STAN, BUGS, JAGS.
        iv. NIMBLE has generic compiler. As far as us building a community, there is space for us to engage with them to extend NIMBLE
        v. Engage them in a partnership or collaboration where their name is on a paper with us will build good will moving forward
        vi. Shouldn't be hard since we are already looped into the community
        vii. A lot of pre-built stuff for observational issue (e.g., dealing with count data for beetle challenge).
        viii. Wouldn't be hard to loop them in to what we are doing.
        ix. Do we want to think about a CI proposal? For funded time?
            1. NASA has an RFP open - due in mid-January
        x. If we want to bring in non-ecologist, a lot of computer scientists interested in tensor flow or caris. Would like to see these other examples for the Challenge.
            1. This is solved in JAGS
            2. This is solved in NIMBLE
            3. This is solved in Greta
            4. Doing a work out using the different tools could be part of the proposal
    k. Mike more interested spinning up forecasts for forecast challenge than writing a methods proposal. But good to put it on the Agenda for January. Get a team from Methods on the line (Leah and other stats people). Go back to what has

been written in the past. What are the methodological issues that are more common for us than for most of the stats community. Think about if there is a proposal. Schedule this with Chris and Perry to see if they are interested in doing this in NIMBLE.

   i. Have to be specific about the use cases because it may be things that NIMBLE hasn't explored (e.g., having NA on the end)
   ii. Take posteriors from one step and feed in as priors for the next. Libraries that feed in as parametric as well as non-parametric. Instead of dnorm have a dkde (kernel density estimator).
   iii. General framework is there. Want to define the use cases and work through the use cases.
   iv. Jody will work with Chris, Perry, Leah, Carl, Mike, Quinn, etc to find time for a call next January.

5. NEON Ecological Forecast Challenge CI Update
   a. Had a day without any error messages in the logs!
   b. Caught issue with flux data. It was finding new data that wasn't actually there.
   c. Only remaining thing is working with John Foster on how to roll out the Ticks Challenge using 2019 data.
   d. Also need to get the null forecast for ½ hour terrestrial. Need to shift from JAGS-focused state space model to sequential DA approach.
   e. If Mike is spending winter break working on models. Is there a way to drop them into the communal CI instead of running them on his server and having to submit every day/month. If Mike creates R script that does it all that includes loading all the packages he needs, then Quinn/Carl can put it in a cron job.
      i. Fork the null forecast and create persistence forecast based on null forecast so containers are all the same as the null will work.
      ii. Tell Quinn/Carl what schedule to run it on.
      iii. Think of it as dropping it onto GitHub with some metadata. If it is simple/common then it will automatically pick it up.
      iv. Could do a simple push to the GitHub repo, they could pull it down and it will update forecast workflow without having to pull a whole new container just because you change one thing in one file. Work with the R Studio container in the CI so you know what it has available.
      v. Prioritize the half hour flux example to serve as the null to put forward.
      vi. Katie has some code to calculate a simple null and a simple persistence so should be able to work with Quin to get those online.
   f. Carl/Quinn to work on wiki on GitHub page so if other people want to spin up from scratch it would have the info they need to know.

6. Forecasting Workflow Updates
   a. Look through the final steps of the Workflow to identify what is covered in the Forecast Challenge and what will be good to keep in mind as a need for the future

b. Put this on the next call