March 6, 2020 CI Working Group Call

Attendees: Carl Boettiger, Alexey Shiklamonov, Jody Peters, Quinn Thomas

Agenda:

1. Methods Group is wondering about having joint CI/Methods calls instead of separate calls. Fri at 3-4 eastern is what we are doing for Methods and for half of CI calls. Had 3-4 pm on Tues for a couple of calls (Feb 11, Mar 31) for CI to have a couple of options for Kenton since he is not available on Fri at 3-4. Jake Zwart can't make Tuesdays 3-4. Current schedule: Mar 6 - CI; Mar 13 - Methods, Mar 31 - CI (Tues), Apr 3- Methods, May 1 - CI, May 8 - Methods
    - Yes! We'll go with Fridays. Jody will work on the schedule
    - Think about working on specific topics
2. Standards:
    - The Outputs and Archives are highest priority
    - Currently, Output standards is being handled primarily by the overlapping Standards group and is fairly far along.
        - Follow up about EML/metadata discussion from Feb 25 Forecasting Standards Call (see notes here:
    - Archives currently remains primarily with CI.
        - Updates from Bryce about archiving model simulation on DataOne:
        - There was a lot of conversation about DOI options on the Feb 11 call. Do we need to keep this discussion going?
            - Carl had a number of suggestions especially about thinking about content-based identifiers and gives some example on this Slack conversation [LINK REMOVED]
    - Quinn has made progress working on the Standards going off of Carl's example. He developed a number of scenarios that we talked through in the following notes. Quinn will make some revisions and share updates in the next couple of weeks as we get closer to the next Forecasting Standards call on March 24.
        - Used Carl's rmd example. Took ensemble-based approach (has most things that will be saved). Created model like Carl did with process noise
        - Forecasting issue time - lets you know what is the time horizon
        - In Carl's code - subproject ID. If you launch automated iterative forecast that gets an ID and each forecast gets a subID which correspond with forecast issue time. forecast Issue time maps on to the subID
        - Quinn added a data assimilation column - it could be useful to know if there is a big period of training. Was data available and used in the model at this time point is what this is a flag for. This could go into the next discussion on identifiers. If you have provenance based statement - this forecast project ID cam from this data assimilation product or raw data. We'll come back to how to handle this. If you have 2 variables in a forecast and they come from different data sources, how to handle this is something to discuss.

- If you have a time series - then there is a time summary column. Gives an example if you have confidence intervals.
- More dimensions - if you have Lotka Volterra model. Has long and wide combined. The observation is a time point and then have multiple states at that time point. The pure long format is manipulation friendly but not metadata friendly. The column names are not informational they say "variable". Quinn thinks this long/wide format that he has used with the Lotka Volterra is the way to go. You could convert to true long format, but he doesn't like it agreeing with what
- Multiple species, with multiple depths (spatial measure), ensemble. Time-space, ensemble. But if people do space with x,y then we recommend that they go to netcdf format. If you have to go to netcdf, you can reduce the size by compressing it.
- Compression look for repeats/de-duplicate which
- Quinn also has netcdf example
- Want to nail the controlled library - this can catch alot of the things the Theory group wants to have
- How much do we want to push the EML standard or put required things in the Methods?
    - Right now EML has the option for a methods section in which people can write whatever they want
    - The far opposite of that is an EFI method requirement which says you have to use our methods and your method section is pre-built for you.
    - Possibly what we want is a blend of that.
    - We want - what is your forecast time step. Did you use DA? If DA, what kind of methods? How many parameters did you calibrate? What is your model name? Et cetera. These would be easy things to force people to fill out. They are not hard things to fill out
    - Carl's intuition for 1st pass - do it as Quinn has shown in his example.
    - Create the method.rmd that people fill out. This would be the best way to move forward now.
- Quinn - then created the metadata for his actual forecast to test it out
    - Kalman ensembles are not associated with depth. Do we create a generic depth that is NA that you can put things in that don't have a depth? If one state variable is depth of water column. That won't be associated with depth, it will be a single value.
    - Are we restricted to a single csv file? What if we have 2 one with state variables and the other with parameters?
    - What is best for archiving? Separate EMLI? You can document multiple tables in EML.
    - Recommend that if you have variables that are 0 and 1 dimension (in Quinn's example) or where variables are 1 and 2 dimension

        - Quinn's example can be easily read in and manipulated while netcdf takes a bit more to generate and to analyze
        - One thing with the netcdf that we haven't been done that we can do. Netcdf carries around alot of the metadata that could be extracted automatically for an EML. But Carl hasn't played around with that to get it to work.
        - There is a lot that is compelling to using netcdf. So could create a wrapper that converts csv to netcdf
        - Because netcdf carries its own metadata, it is very powerful.
    - Best thing we can do - decide what needs to be in a csv and netcdf file.
        - Time needs to be in a specific format
        - If there is a spatial dimension - do we need to say it needs to be defined as a specific term. But then we might be reinventing spatial raster formats
        - x,y,z are the coordinates. There are spatial options to describe that as lat/long, or specific spatial coordinates
        - Is it worth saying if it is 1 dimension that it is x, if it is 2 dimensions it is x,y, and if 3 dimensions say it is x,y,z. Not sure if we get data with multiple dimensions will need to have all the spatial information that corresponds with those dimensions
        - 2 modes - you do it in csv, call it depth and a human has to read metadata to figure it out. If you have spatial data, you have to have all the coordinate details
    - We want to be able to interpret the forecasts as they come in focusing on things that let us figure out what is different between forecasts
        - Do we create a set of rules? Or embed in an EML format?
        - Have classes of columns (ID things, spatial details, etc). If we can agree on 4-5 categories for each of these things and give the categories specific identifiers
    - Quinn to package up what he has developed and will share with the group for review/comment

- Auto archiving and how you get identifiers is another topic
    - When is a unique ID minted for a new forecast?
        - Something that would get a unique ID is after you have done the coding, have the model working, launch the iterative forecast and have put it on server and it starts running.
        - Then each subsequent iteration gets unique sub-identifiers (this is the forecast of Apr 14 of this set of code/data/launch from the unique ID)
        - Anything that requires human intervention or changes becomes new DOI. If you change anything by hand it is a new DOI

- Project ID is a manual and subproject is anything automated - need standard/best practices for this
    - Sub ID is unique and guaranteed to be unique across all identifiers and sub IDs are nested under a Project ID
- For paper citations - would want to cite the Project ID - it is a unique ID, but is not automated
- Tie project IDs like git commits (or MD5 checksum, etc.). If you have to turn it on and back off again then it doesn't get a new project ID)
- Changes in input files - if weather forecasts get pulled in new forecasts every day. Then doesn't get new ID. But if workflow and parameters for the model change, then get a new ID.
- How does this intersect with more long-term archiving? DOI's are project ID level. Can our current archiving systems handle the subproject IDs? What archives are best? Or do we make recommendations to archives to handle archives better?
- At U of IL - made a deal to have a single DOI for a database.
    - If you reference a specific run - how do they look at the DOI?
    - Don't think we need all the infrastructure behind the Sub ID that we have for DOIs. Leave the SubID internal to the archive
    - Need to have an archiving system that is able to continuously get updated while having the same DOI
- Rob - looked at Dryad
    - http://wiki.datadryad.org/Track_Version_Changes
    - http://wiki.datadryad.org/DOI_Usage
- If you update your object on a repository, it gets a timestamp. Can you not link a package to that? You can go back and change a forecast, but that would also change the timestamp on the publishing of the repository.
- The effort to decide to launch a forecast is going to be a manual process.
- It is more important to have best practices that we encourage people to follow than having code/script

    - Input Standards. Lower priority but Jody is keeping this on the Agenda as a reminder to work on this.

3. Methods is also continuing to work on the forecasting tools EFI Task View list of resources and would love help from CI on areas of overlap (which is pretty much everything except the hard-core stats part). They are planning on cleaning the task view up into a more polished doc over a series of 3-4 blog posts. First blog (led by Jake): overview of 9 areas, details on Workflows, Reproducibility,

- Data ingest, cleaning/harmonization, management
- Visualization, decision support, UI
- Stats/modeling (general and data assimilation / data fusion specifically), uncertainty quantification and propagation

Want this done decently ahead of the RCN so we can also **ID the gaps specific to the RCN forecasting challenge** and maybe focus some development effort on some RCN specific tools (e.g. weather forecast scripts)