February 11, 2020 CI Working Group Call

Attendees: Carl Boettiger, Rob Kooper, Bryce Mecum, Quinn Thomas, Jody Peters

Summary: In collaboration with the Theory Group on the Forecasting Standards call, this group will continue to define what a forecast archive should look and what should go into archived forecast packages that get saved on any of the archiving platforms. The next steps, which we will discuss more on the Feb 25 Forecasting Standards call, are to use Carl Boettiger's simple example as a starting point to test creating metadata for a couple of operational forecasts (forecasts with NEON data would be good test cases as we prepare for the RCN meeting). We will then want to compare across the metadata to see if all desired information to find forecasts of interests (e.g., forecasts for specific species or locations or using certain models) and to be able to compare across models is available.
Bryce Mecum offered to look for examples of how people have archived model simulations rather than forecasts from the DataOne archive.
The group also talked about how to handle DOIs for forecasts. We discussed wanting to make forecasts searchable with DOIS, but not needing DOIs for every daily updated forecast output. Other options discussed were UUIDs that DataOne uses as well as the DOI versioning that Zenodo employs. This will continue to be a topic of discussion.

Agenda:

1. Updates:
    - On Dec 16, Alexey submitted input to the NSF RFI on Data-Focused CI Needed to Support Future Data-Intensive Science and Engineering Research
        - Here is the link to Dear Colleague Letter for the RFI: https://www.nsf.gov/pubs/2020/nsf20015/nsf20015.pdf
    - EFI-RCN Boulder Meeting May 12-14
        - Everyone should apply! Deadline is this Friday, Feb 14. Here is the website with info and the link to apply

2. Archiving Platforms - Want to get something pulled together before the EFI-RCN May meeting. **Here are ideas from previous calls**.
    - Continue Slack conversation about archiving platforms
    - Develop a list of needs for platform(s)
        - How well does the platform support the guidelines for archiving
        - Long term viability. Forecast that won't go away
        - Machine pushable/writeable. Automate workflow
        - Accommodate frequent updates (without creating unique DOIs each time)
        - Want it to be discoverable (e.g., new DOI issue)
        - Use this list of needs and how they relate to each platform would work for the blog. Plus with the script examples

- Have folks familiar with each platform OSF (Alexey), Zenodo (Ethan), DataONE (Bryce/Matt) summarize how each platform fits the needs
- Create blog post summarizing pros and cons
- Throw together basic script to show example of automatic upload. Alexey could put this together for OSF. Check with Ethan to see - he probably already has this for Zenodo.
- **Discussion from 2-11-20 call**
- Carl: Risk getting sucked into goals for functions of different platforms and helping people choose. Platforms have a lot more in common then they are different. Our role is about what should a forecast archive look like on any of these platforms.
    - Don't want us to get hung up on the micro features of the platforms
- What would go into an archive package?
    - Operationalize what you expect to do with the forecast that really matters.
    - Have an automated tool that gives a score between multiple forecasts
    - Have to work backward from 2 comparable forecasts. What would need to be capture from both of them that is automated that makes them comparable?
- Ethan has developed code for automation creation of archives on Zenodo.
- Defining standards and pointing people to resources. Example of one on one platform.
- Would love to see NEON based examples.
- Goal going into May meeting is having some preliminary NEON forecasts - use them as a way to workshop this archiving structure
- Hackathon in mid-RCN years - would be an opportunity to develop tools that are widely useable.
- Go into the RCN meeting - we want to have an idea of what an archive package would look like and wedge it into existing forecasts.
- Provide a review of what that package would look like and try shoving some data into it.
- Ideas of what would go into forecast archive package
    - Metadata file
        - Parameters that go into the model that the model needs to run the forecast
        - Descriptors of the forecast
        - Class of model
        - What date it was run (timestamp)
        - Version of model
        - If forecast is reproducible with containers
        - Is data included and what kind of data? And what do the data go into (model calibration - could have been done years ago; data assimilated into model; data used to evaluate model when the data comes in)

- We are close to the Metadata. Now we need folks to wedge forecasts into these Metadata
  - Are there things we need? Do we have the correct information/enough information that we need?
  - Will need to iterate on real examples to help answer these questions.
- Need to define the set of functions we need to run on the metadata
- From Carl's example - Output data is the Metadata that travels with it. His example doesn't have uncertainty included. Not sure if we have a good system for uncertainty.
  - Would want it to be an attribute. List the uncertainty types included in forecasts so people would know what you did
  - Could be an attribute on a table (one column that is point estimate, other column is for interval, and third column is the type of interval). It's possible
  - As for EML and what can be done with Schema it is limitless. The hard part is authoring the EML. The tools that Carl has been building has been working on to generate EML automatically
  - Leveraging the EML tools and adapting is good.
  - In EML (XML Schema side) you can't have things that are required vs optional. Would need to create an R package to do this
  - Are there examples of how people have archived model simulations rather than forecasts. There are 100,000 examples, so we could probably find some examples
    - Bryce can look at this and can send some examples if he finds them
  - If R package or web form doesn't provide an easy way to set up the XML, people won't add the complexity
  - Carl's example is simple. But it points folks to how to interpret data file that travels. Need to define use case for the forecasts. EML is a good starting point for this. Discovery of the data is one example (want to know the species or the location forecasting that has happened vs finding forecasts that use a particular approach vs want computer to read EML and re-run the forecast)
  - Need to in 30 years create the figure that weather forecasters can do now - which is to show that our skills have gone up over time. Need the metadata to do that in 30 years.
  - Identify infrastructure that is specific to forecasting, then show things such as DOI is not what we want for each new iterative forecast.
  - Either we identify the problems and solutions or identify the problems and find the people who can work on that problem.
  - Identify forecast-specific issues. Understand that some of the general CI issues won't be solved.
  - EML is not too far off from allowing us to have the metadata needed about the forecasts

- Solving the DOI issue is important. Put this on the list of things to work through. What is the best solution that is not burning through DOIs?
- Are there other persistent identifiers?
    - A group could mint their own
    - Are there are examples where a DOI wasn't an appropriate tool?
        - DataOne is in the million of identifier space. Lots of those do use DOIs. The DOIs are the human - found identifier
        - UUID - probabilistically chosen random globally unique identifier. This is an opaque identifier.
        - If you launch forecast for a system - that might get a DOI, but every day's forecasts would get the UUID
        - DataOne has this model: have a series of containers that are named the same. Example: doing a forecast that changes over time. Give one DOI that points to the containers for the forecasts that may change. The raw model output gets opaque identifiers. This fulfills the demand of the user. But the problem is getting back to the bytes people want. Bryce's example - writing koala data for a journal. Getting to the specific data/output/bytes that need to be cited can be difficult
    - Zenodo uses DOI versioning
        - Have 1 DOI which points to the main element. Then every day have a new version of the dataset, so have a new DOI for each day.
        - Zenodo has unlimited number of DOIs
        - But versioning in Zenodo is different from Bryce's example f subcomponents. Upload foreacsts have output files, metadata, input files. Bundle all that into Zenodo and give a DOI. Later using new data can re-run the forecast in Zenodo and rebundle. If you want to refer to a particular portion within the model you don't have the UUID to find the info.
        - Two things that might happen: 1) I am using the same code, but the input data has changed. 2) Or perhaps the input data hasn't changed but the code has changed. At a fine scale we don't have the infrastructure or haven't thought of how to use the infrastructure to organize these two types of situations.
        - Forecast unit - expect to have the same model and expect to have new data coming in.
        - If you have a new/change the model then that would get a new DOI
        - Do we define a forecast as all the occurrence of the model and the window of data that goes into machine and

window of data that comes out? Or a subset of those pieces?
    - Example: Forecast is to propagate the average of the data into the future. As you get more data the forecast will change
        - If download the data from NEON today and download a month from today and run average algorithm, it should produce the same result.
        - This is sort of a continuation of the data because the model doesn't change. If you change parameters then you have changes and that would be a new version
        - If you re-calibrate the model to a new dataset, then you would relaunch the forecast and that would be a new identifier.
    - What do we do for a continuous run of the models?
        - If the model equations haven't changed and you have been iteratively ingesting new data, then it is the same forecast
        - Is there a right/wrong time to mint a DOI - this is a different question, then can we describe what we have done with enough provenance.

3. Input Standards - we didn't talk about this today. The Outputs and Archives are higher priority, but Jody is keeping this on the Agenda as a reminder to work on this.