March 17, 2026 Joint Methods & CI Working Group Call

Attendees: Dave Durden, Brittany Barker, Josh Cullen, Josie Hughes, Jody Peters, Jake Zwart, Will Hammond, Marc Cotnoir

Agenda:

1. EFI2026 Conference - Abstracts are due April 15
    a. See details here: https://bit.ly/efi2026

2. Show and Tell: Methods for delivering forecasts. The group was interested in discussing differences in sharing forecasts, e.g., R Shiny, other methods with Leaflet and Java servers, etc. Are there tools that are user-friendly and aren't slow? Are there suggestions for improving the speed adn scalability of R Shiny apps or other open sources tools for spatial delivery? What has worked for others? Where are people running into issues?
    a. From Ethan in Slack: This is raster data I'm guessing? If so what's the rough size and are you just visualizing individual rasters or doing more interactive work with the data. I'm guessing you all are well beyond us already, but we've had some success with modestly large (5-100 GB) underlying imagery through shiny/web apps. Our stack is pretty dated at this point but the key component is tiling the rasters to only load the data you need and only at the resolution needed for the level of zoom. Happy to chat more if useful or just follow along to see what we should upgrade our stack to.
    b. From Josh via email: Here's a proof-of-concept dynamic ocean management tool I put together while developing the cyberinfrastructure to be used in a larger project. At the moment, I have it included as an embedded Shiny app in a Quarto website, where I've stored all environmental variables that went into model predictions as a cloud-native Zarr file in a Google Cloud Storage bucket (since these datasets get really big really quickly) while reading in the daily model prediction rasters directly from my GitHub repo (for now). I'm working to speed up the load time of the Shiny app, but I think this is mainly due to requiring the use of Python packages to handle the Zarr data and pass to R via {reticulate}, which requires installing the Python packages upon cold-start of the Shiny app. I'm trying to see if I can make these load times faster.
    However, I've played around with the {mapgl} R package for interactive maps (in place of Leaflet) and will likely pivot to this tool instead since it definitely seems to be faster and more responsive (built using the Web GL JavaScript library that's optimized for speed and scalability). I've only really used this package for handling large vector data, but will test it out soon for raster data as well. Alternatively, it looks like the Python 'anymap-ts' package that was just released by Qiusheng Wu (also built on top of Web GL) provides even more nice-to-have options. So I might explore this route as well.
    c. Will H., any updates on the visualization for the fire recovery spatial forecast challenge?
    d. Discussion on the call
        i. From Josh:
        ii. NOAA has big push to move to the cloud
        iii. Dealing with ever growing scale of data
        iv. https://github.com/joshcullen/CEG_operationalization
        v. Uses quarto - ups in products and has badges to show if the products are up and running
        vi. People originally wanted static maps

vii. Also has embedded Shiny map - used leaflet, has layers to toggle on/off, has environmental data an dmodel predictions.
viii. Model predictions had been stored in GitHub repo, but 1000 file limit and hit that quickly. New approach - storing a zarr file (cloud version of netcdf) in Google bucket. Have python script with xarray package
ix. Use GitHub actions and the zar file so doesn't need to be committed to GitHub
x. Spatial resolution of rasters - Brittany has issues with her rasters loading. Josh's raster resoultion are 0.25 degree.
xi. From previous work used county level data to look at risk of avian influenza. Would take 10 seconds for each layer.
xii. Thinks {mapgl} might be good. Could submit an issue to reach out to Kyle Walker who runs mapgl and if the request isn't too crazy, could be added if there was something in leaflet that is not in mapgl that are wished for
xiii. Didn't find a good R package for working with the Zarr file. So used xarray from python instead. But then need to hand things over between
xiv. Used r reticulate package to pullin tings from the Zarr file, then once in python format, then exported to create terras raster to create the visualization in Shiny
xv. From Josie:
xvi. Evaluating bird species experts to look at bird distribution while being flexible to provide feedback on spatial SDMs. https://github.com/LandSciTech/sdmEvaluationTool/tree/main
xvii. Has initial prototype that deploys on individual computers.
xviii. Use R Shiny leaflet and geotiffs for people to toggle on/off things
xix. Feedback allows people select evaluation subunits on a fixed grid
xx. Issue on speed is not so much on the rasters, as it is the mechanism for people to review and provide feedback on the input data which is more of a point data set with lots of points. Loading the points and interacting with them is too tricky. So the first pass has been to rasterize the point data
xxi. Much of the work has been focused on the process of figuring out the right questions to ask the evaluators
xxii. Really exciting tool for connecting modelers and land managers
xxiii. Core data sources - acoustic recording units, federal monitoring programs, ad hoc deployment of recorders, point count data. Wide range of data all compiled
xxiv. You can cluster points (like on Zillow or other websites) when zoomed-out, which then separate upon zooming in. It's possible to do in Leaflet, but probably slow if you have a lot of points. Much faster to load using {mapgl} in my experience: https://walker-data.com/mapgl/articles/layers-overview.html#circle-layers

e. From Brittany:
i. https://uspest.org/CAPS/ Has static forecast for 18 invasive insect species and will add some plants
ii. Boxwood blight risk mapping - example of the plant disease Brittany has worked on. This is a Shiny app: https://riskmaps.oregonstate.edu/boxb/
    1. Stakeholders are people doing surveillance for the boxwood blight
iii. Rasters are stored on servers at OSU
iv. One thing Brittany will try is to break the rasters into tiles. Or may try a png file at large scale and then go to a raster when zoomed in

3. April or May call - Leah Johnson and Parul Patil could lead a read-through of their updated manuscript, "Gaussian process forecasting of sparse ecological time series."