Talk proposal: Programming Opportunities for the Global Biodiversity Observation Network

Jean-Michel Lord [1,2,3], Jamie M. Kass[1,4], Andrew Gonzalez [1,2,3], Michael Dales [5,6], Anil Madhavapeddy [5,6]
[1] Group on Earth Observations Biodiversity Observation Network (GEO BON)
[2] Quebec Center for Biodiversity Science, Québec, Canada
[3] McGill University, Québec, Canada
[4] Macroecology Lab, Graduate School of Life Sciences, Tohoku University, Miyagi, Japan
[5] University of Cambridge, UK
[6] Cambridge Conservation Initiative

Actions to reverse the global biodiversity crisis demand a rapid, coordinated response, underpinned by robust, comparable, and actionable information on the status and trends of nature to detect and monitor changes, as well as direct evidence-driven action. While societies have long monitored nature, a renewed commitment is urgently necessary to provide the evidence needed to guide conservation decisions, particularly at the scales required to change outcomes for nature and people inscribed in the Sustainable Development Goals (SDGs) and the targets and goals of the Kunming-Montreal Global Biodiversity Framework's (GBF) Monitoring Framework.

However, our ability to detect and understand large-scale trends is challenging because biodiversity data remains largely fragmented and biased, and also because the approaches used to record, aggregate, and analyse monitoring data to detect trends and attribute causes are varied and not standardized. These challenges slow down progress and make reporting practices inconsistent across projects, organizations, and nations, thereby hindering effective assessment and action. The collection and documentation of biodiversity data has increased exponentially in recent years with a similar growth in the number of databases. However, heterogeneous coverage, protocols, and standards hamper their ready use and interpretation.

An ongoing effort has been the establishment of global Biodiversity Observation Networks (BONs) that are a community of practice, applying a network approach to coordinating biodiversity observation programs and efforts. A BON can be:

1) local, sub-national, national, regional or global in scale of operation and can cover one or multiple aspects of biodiversity;
2) structured around themes, methods, biomes or specific taxonomic groups; and
3) a networking point for people who want to strengthen biodiversity observations in a region.

BONs coordinate the collection, dissemination, and use of national, regional, or thematic biodiversity data, thus acting as both data generators and hubs.

Some[1] of these are organized collectively under the global organization “GEO BON”, which promotes collaboration and coordination among BONs.

However, without a unifying programming platform to consolidate the diverse measurements across BONs, it will remain difficult to use these datasets reliably. For this reason, GEO BON piloted the creation of BON in a Box, an open and collaborative platform that provides a suite of analysis tools and resources, supporting biodiversity analyses and the establishment and operation of BONs. In this talk, we will describe the emerging "BON in a Box Pipelines" module that offers four key functions:

1) **calculating** essential biodiversity variables—that describe the state of nature—through modular and reusable pipelines to enable rapid detection of biodiversity trends;
2) **calculating** biodiversity indicators to track progress towards goals and targets;
3) **enhancing monitoring** by suggesting priority areas where data gaps necessitate further field data collection to improve models (under development); and
4) **sharing** science by transforming manual workflows into automated data-transformation pipelines that can be reproduced or improved by others.

Advances in programming languages and computer systems have a key role in shaping the usability of each of these key functions, which will be the focus of our talk.

Architecture

BON in a Box Pipelines provides an open-source, reproducible platform for users to contribute data and analysis pipelines to calculate indicators that can assess progress towards biodiversity targets. BON in a Box can integrate diverse data and expert knowledge, ranging from Earth observations to occurrence metrics. The platform lowers technical hurdles for non-programming biologists to run these analyses by focusing on their expertise—knowing the local environment and its ecology, as well as assessing the results—rather than the code behind the data transformation.[2]

The BON in a Box Pipelines platform is fueled by a community-contributed and open-source repository of scripts that, once assembled, convert biodiversity data to either essential biodiversity variables (EBVs) or biodiversity indicators. It does not mandate a single programming language, but instead coordinates programs written in different languages (primarily R, Julia, and Python). These pipelines combine both local results (i.e., national, regional) and also previously calculated global results.

Individual Pipelines

Pipelines in BON in a Box transform input data on biodiversity and the environment into valuable output that helps us meet global conservation goals. They can produce EBVs,

[1] Since this is all open source, there are numerous national and subnational initiatives not directly tracked.
[2] See https://ecoevorxiv.org/repository/view/7941 for a preprint on Bon in a Box

indicators, or other products essential for monitoring and decision-making in the context of biodiversity conservation and natural resource management.

Each step in a pipeline is a script that processes input data by cleaning, augmenting, standardizing, or by performing statistical analyses. Importantly, these scripts can be structured and repurposed modularly across varied contexts. For instance, data-cleaning scripts can be applied to multiple pipelines that use similar types of data, enhancing efficiency and consistency in data-processing workflows. An example is a script[3] that cleans species' occurrence records from the Global Biodiversity Information Facility (GBIF).

Pipeline composition

Outputs are stored directly into the filesystem, allowing the pipeline to abstract the specific choice of programming language. A typical pipeline can (and often does) mix R, Python and Julia scripts without interoperability issues. Docker containers are used as the basis for executing the pipeline, with volume mounts to access the inputs and outputs filesystem.

Pipelines can build upon the results of other pipelines, meaning an "inner pipeline" can be included as a step in the "outer pipeline". For example, a species distribution modeling (SDM) pipeline could serve to generate species range map predictions inside a species habitat index pipeline, or it could be embedded in a species richness calculation pipeline that combines individual range predictions.

Opportunities for leveraging computer science

We will discuss several opportunities to further develop new capabilities for BON in a Box by leveraging best practices in computer science.

Opportunity: change detection in code

Individual pipelines are also peer-reviewed to ensure they meet community standards and actually implement the scientific methods they claim to.

https://geo-bon.github.io/bon-in-a-box-pipeline-engine/pipeline_standards.html
https://geo-bon.github.io/bon-in-a-box-pipeline-engine/peer_review.html

While the first round of peer-review is fairly straightforward, whenever scripts are subsequently updated, the current system necessitates a full re-review. In addition to enhancements, updates can be triggered by breaking changes in remote APIs.

We therefore need to consider how to make change detection more of an automated process across often messy datasets. This could be done, for example, by programs that compare system calls between two package versions (does the new package version really change something for the subset of code we are using?), calculate statistical outliers (does the mean

[3] See https://github.com/GEO-BON/bon-in-a-box-pipelines/blob/main/pipelines/SDM/SDM.md

vary over a certain threshold?), or apply more static assertions (do hard-coded checks result in the same values?). The end-goal is to update dependencies without needing to conduct a review, or for a human to review only the part of a pipeline that actually changes before enhancements are merged.

Opportunity: visualisation

The inputs and outputs of a pipeline are typically quite amenable to interactive visualisation, but this is an ad-hoc process in the current version of the application. Each data type has a visualizer (i.e., function to plot or map data) that is coded directly in the frontend. Since the data types are open-ended, the need arises to allow pipeline contributors to choose the most relevant visualisation for their input and output data. Adopting client-side notebooks to encourage the modular bundling of a data library (perhaps in Webassembly) with associated user-interface elements would help the developers of BONs explore their data and analysis products. Flexibly embedding these in diverse development environments would also assist with the exploratory scientific process.

Opportunity: dynamic dataflow in pipelines

Pipelines are currently built manually in the pipeline editor, then saved as static dataflow graphs in JSON format, which is simple and interpretable. However, more complex pipelines will inevitably benefit from data-dependent control flows, in other words, pipelines that adjust themselves to the input data.

The most common example of this is to compose pipelines to convert custom data types into the types expected by the initial steps without affecting the (peer-reviewed) core of the pipeline. A more complex example could be detecting a corrupted input data store and branching off a repair script. Detecting and proposing these adjustments for user approval would reduce the number of data transformations to apply beforehand, and possibly suggest techniques not previously known by the user. Semantic modeling[4] or other suitable approaches could be used.

Opportunity: chains of trust for federation

We envision the creation of multiple instances of BON in a Box, for example, local instances on researchers' laptops, central BON instances on servers, and reporting processes for conventions.

BON in a Box instances are currently public, and the only way to restrict access to the processed data and results is to lock out the whole node via server-side authentication. There are good reasons to do so, since disclosure of sensitive information can endanger vulnerable species. However, when building a federated system where the instances report to a BON, which in turn reports to organisations like the Convention on Biological Diversity, a system must be in place to secure the flow of information. The data flagged as sensitive should restrict access to all derived data, viewable only to authenticated users, along the chain of trust.

---

[4] Annotations about data meaning and relations can be used to dynamically create data transformation workflows. Example at https://github.com/integratedmodelling/klab

There are several emerging technologies to support this use case, such as transparency logs for TLS certificates, the ATProto decentralized protocol for signed data repositories that drives the Bluesky social network, and labelled information-flow control. However, these need to be deployed into biodiversity networks with extreme care due to the high stakes if sensitive data were to leak; poachers and other malicious actors are quick to act and the consequences for rare species cannot be easily remedied retrospectively.

In summary, our talk will lay out the overall state of global biodiversity observations, and serve as a call to motivate computer scientists to contribute to the digital structures behind them. We believe that a wide array of contributions from programming languages, distributed systems, and security all have a role to play in meeting the urgent challenge to conserve biodiversity in the coming years.