Community-developed data exploration

A developing climate and GFDL community data exploration and analysis toolset

Aparna Radhakrishnan
Ciheim Brown
Chris Blanton

GFDL Informal webinar, March 12th, 2024

SAIC

Agenda

- Motivation (Chris) + a Role play activity (all presenters)
- Community acknowledgement and intake-esm overview (Aparna)
- Documentation and Catalog generation (Ciheim)
- Data exploration and FRE workflow connections (Chris B)

Acknowledgment:

- Bennett Chang, Jessica Liptak, Wenhao Dong, John Krasting, Lori Sentman, Raphael Dussin, Ray Menzel, Eric Stofferahn, Jeff Durachta, Chan Wilson.

Housekeeping:

- We will stop for 1 or 2 clarification questions after every section. The meeting will be moderated. Use this doc for Q&A. GFDL DEIA Code of Conduct to be adhered to at all times.

Motivation…

Process-Oriented Diagnostics

Principles, Practice, Community Development, and Common Standards

J. David Neelin, John P. Krasting, Aparna Radhakrishnan, Jessica Liptak, Thomas Jackson, Yi Ming, Wenhao Dong, Andrew Gettelman, Danielle R. Coleman, Eric D. Maloney, Allison A. Wing, Yi-Hung Kuo, Fiaz Ahmed, Paul Ullrich, Cecilia M. Bitz, Richard B. Neale, Ana Ordonez, and Elizabeth A. Maroon

KEYWORDS: Atmosphere-ocean interaction; Clouds; Hurricanes/typhoons; Hydrologic cycle; Model evaluation/performance; Interannual variability

ABSTRACT: Process-oriented diagnostics (PODs) aim to provide feedback for model developers through model analysis based on physical hypotheses. However, the step from a diagnostic based on relationships among variables, even when hypothesis driven, to specific guidance for revising model formulation or parameterizations can be substantial. The POD may provide more information than a purely performance-based metric, but a gap between POD principles and providing actionable information for specific model revisions can remain. Furthermore, in coordinating diagnostics development, there is a trade-off between freedom for the developer, aiming to capture innovation, and near-term utility to the modeling center. Best practices that allow for the former, while conforming to specifications that aid the latter, are important for community diagnostics development that leads to tangible model improvements. Promising directions to close the gap between principles and practice include the interaction of PODs with perturbed physics experiments and with more quantitative process models as well as the inclusion of personnel from modeling centers in diagnostics development groups for immediate feedback during climate model revisions. Examples are provided, along with best-practice recommendations, based on practical experience from the NOAA Model Diagnostics Task Force (MDTF). Common standards for metrics and diagnostics that have arisen from a collaboration between the MDTF and the Department of Energy's Coordinated Model Evaluation Capability are advocated as a means of uniting community diagnostics efforts.

In the quest to improve climate and weather model simulations, process-oriented diagnostics (PODs) have been advocated as a means of providing more information to model developers beyond performance-based metrics. A POD (Eyring et al. 2005; Sperber and Waliser 2008; Maloney et al. 2014; Kim et al. 2014; Eyring et al. 2019; Maloney et al. 2019) characterizes a physical process that is hypothesized to be related to the ability of a model to simulate an observed phenomenon. Evaluating a candidate model version against observations analyzed with such a POD can, in principle, give insight into whether a particular process is being well represented, focus model improvement on specific processes, and identify gaps in the understanding of phenomena. However, moving from principles to practice is a nontrivial step that requires thoughtful implementation. Here we draw on experience with the NOAA Model Diagnostics Task Force (MDTF) to provide best-practice recommendations for entraining diagnostics from a broad scientific community to make them available to model developers.

Developing a suitably comprehensive and useful package of diagnostics to enable effective climate model validation is a challenge. The set of phenomena that a numerical weather, climate, or Earth system model (ESM) is expected to capture is continually expanding, and the observational datasets to which the model can be compared continue to be expanded and improved (Teixeira et al. 2014; Eyring et al. 2016a). A model development team normally has a set of diagnostics from prior phases of model development but has limited resources for maintenance and further development of those diagnostics. Legacy diagnostics packages can quickly become outdated as new observational datasets are developed and are often associated with a particular developer who may have moved on from the organization. It can thus be highly advantageous to have a mechanism by which diagnostics development in a wider set of research groups can be brought into a coherent framework for use by the model development group.

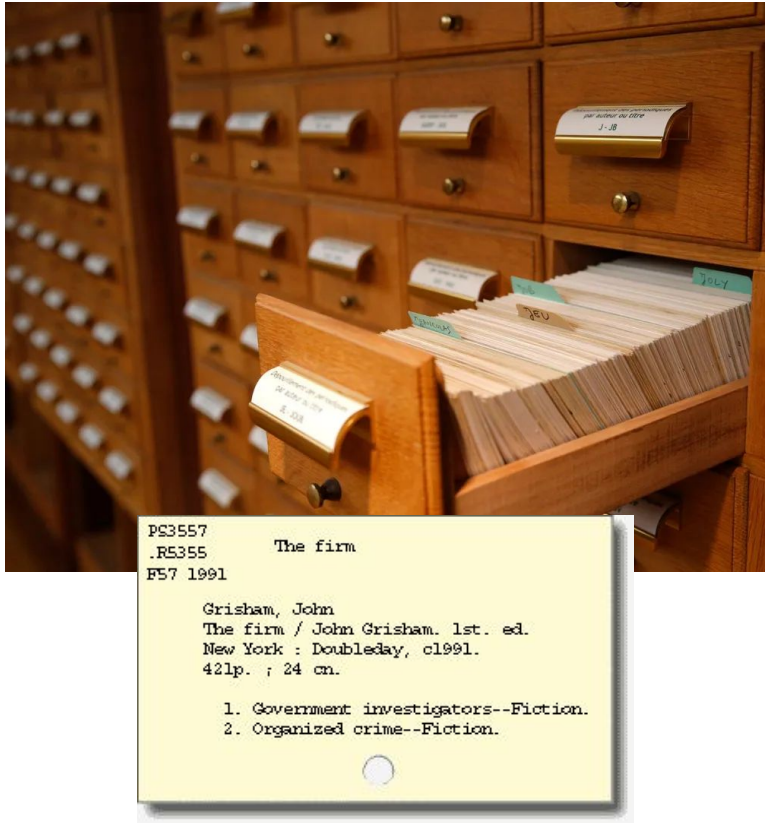
What are catalogs?

- If you managed to “explore” big data archives, to get to a usable outcome: you’ve likely incorporated catalogs.

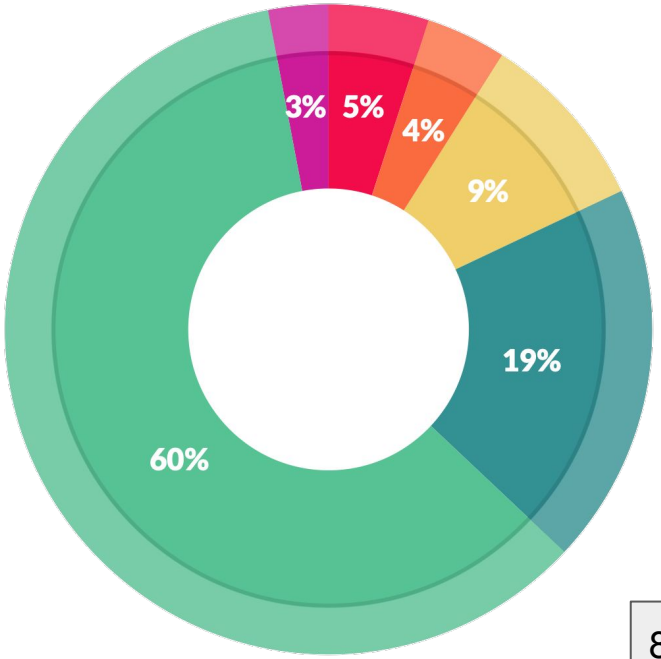
- **How do we go from usable to:**
  memorable, REUSABLE and INTEROPERABLE?

  Opportunity to leverage CMIP-like conventions, extend and overload it for our use.

- “data catalog”
  == community-developed data exploration tools



The need for data exploration



What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

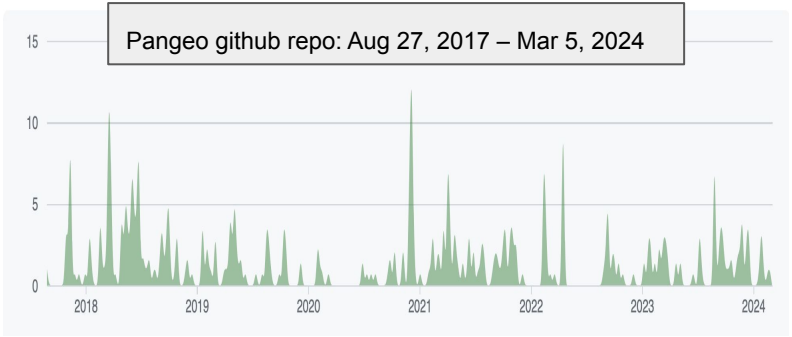
80% of their time on preparing and managing data for analysis.

Ref: How do data scientists spend their time?
Crowdflower Data Science Report (2016)

Acknowledging community collaborations

PANGEO

A community platform for Big Data geoscience

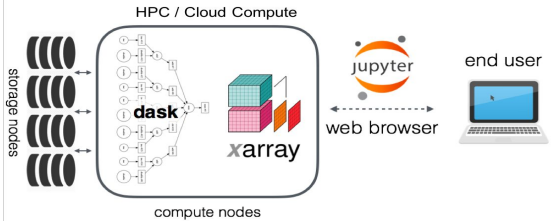


CMIP6 Hackathon (2019)

CMIP

select-OBS

How can search for data?
How can I load the data in my multi-model analysis?



[2020-2021]

Informal Pangeo/ESGF Cloud Data working group

Columbia University

Centre for Environmental Data Analysis

SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

SAIC UCAR

PRIVATE SECTOR

NCAR

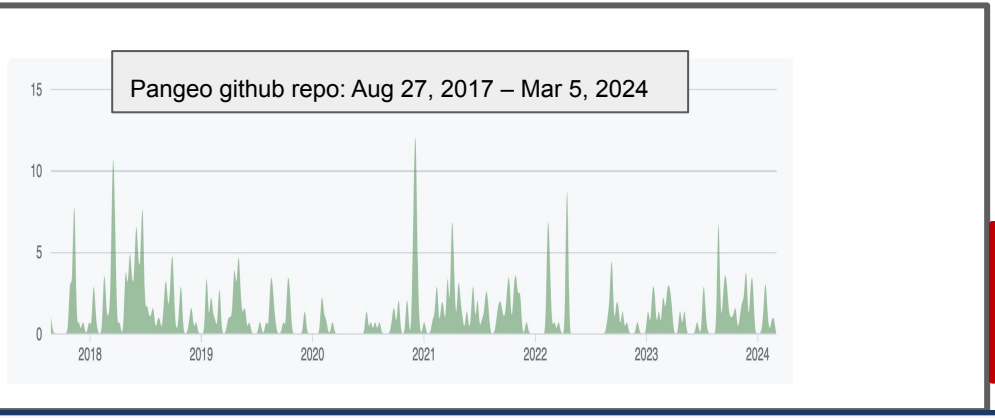
DKRZ
DEUTSCHES KLIMARECHENZENTRUM

NOAA Model Diagnostics Task Force

Acknowledging community collaborations

PANGEO

A community platform for Big Data geoscience



CMIP6 Hackathon (2019)

CMIP

select-OBS

How can search for data?
How can I load the data in my multi-model analysis?

Intake-esm was introduced

[2020-2021]

Informal Pangeo/ESGF Cloud Data working group

Centre for Environmental Data Analysis

SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

SAIC UCAR

PRIVATE SECTOR

NCAR

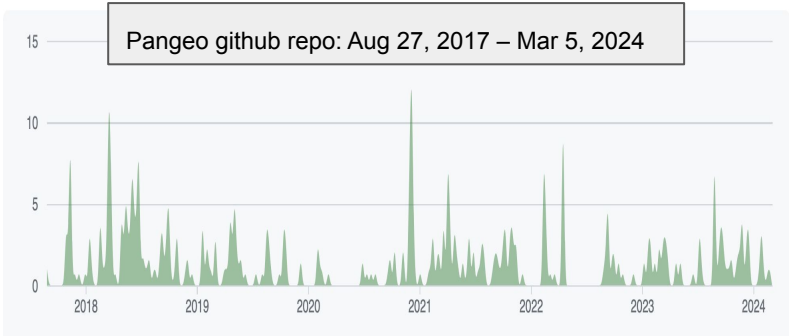
DKRZ

NOAA Model Diagnostics Task Force

Acknowledging community collaborations

PANGEO

A community platform for Big Data geoscience

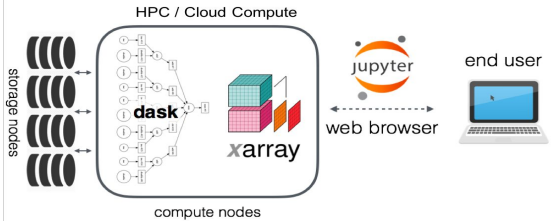


CMIP6 Hackathon (2019)

CMIP

select-OBS

How can search for data?
How can I load the data in my multi-model analysis?



[2020-2021]

Informal Pangeo/ESGF Cloud Data working group

Columbia University

Centre for Environmental Data Analysis

SCIENCE AND TECHNOLOGY FACILITIES COUNCIL
NATURAL ENVIRONMENT RESEARCH COUNCIL

SAIC UCAR

PRIVATE SECTOR

NCAR

DKRZ
DEUTSCHES KLIMARECHENZENTRUM

NOAA Model Diagnostics Task Force

Data catalogs? intake-esm?

Catalog Specification

- what we expect to find inside and how to open the “datasets”/objects?
- Provides metadata about the catalog
- Identifies how multiple files can be aggregated into a single “dataset”
- Extensible metadata
- Single JSON File

Catalog

- Tells us more about the data collection
  - Path to the files (objects), and associated metadata.
- User-defined granularity
- CSV File

Intake-esm: Opens possibilities to QUERY and ANALYZE

- Provides a pythonic way to “query” for information about data collections.
- Loads the results in an xarray dataset object

Data catalogs? intake-esm?


```
{
  "esmcat_version": "0.1.0",
  "id": "sample",
  "description": "This is a very basic sample ESM catalog.",
  "catalog_file": "sample_catalog.csv",
  "attributes": [
    {
      colum_name": "experiment_id",
      "vocabulary":
"https://raw.githubusercontent.com/WCRP-CMIP/CMIP6_CVs/
master/CMIP6_CV.json"
    } ,
    {
      "column_name": "variable_id",
      "vocabulary": ""
    },
    {
      "column_name": "path",
      "vocabulary": ""
    }
  ],
"assets": {
    "column_name": "path",
    "format": "netcdf"
  }
}
```


experiment_id,variable_id, path
cmdev-test,ts,tsfilename.1900.nc
cmdev-test,ts,tsfilename.1904.nc
cmdev-test,ts,tsfilename.1905.nc
cmdev-test,thetao,thetaofilename.1900.nc
cmdev-test,thetao,thetaoilename.1904.nc
cmdev-test,thetao,thetaofilename.1905.nc

```
col = intake.open_esm_datastore(path_to_catalog_specification
```

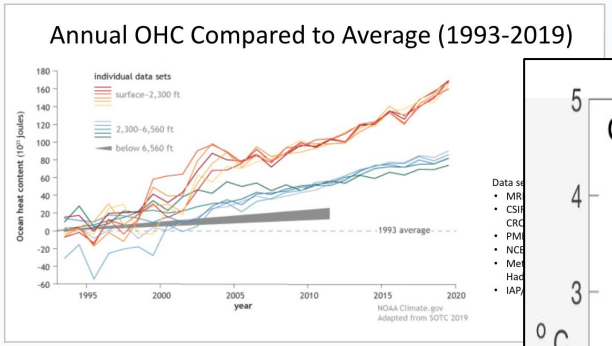
catalog with 2 dataset(s) from 6 asset(s):

```
exp_filter = [omdev-test]
variable_id_filter = "ts"
cat = col.search(experiment_id=exp_filter,
                 variable_id=variable_id_filter)
```

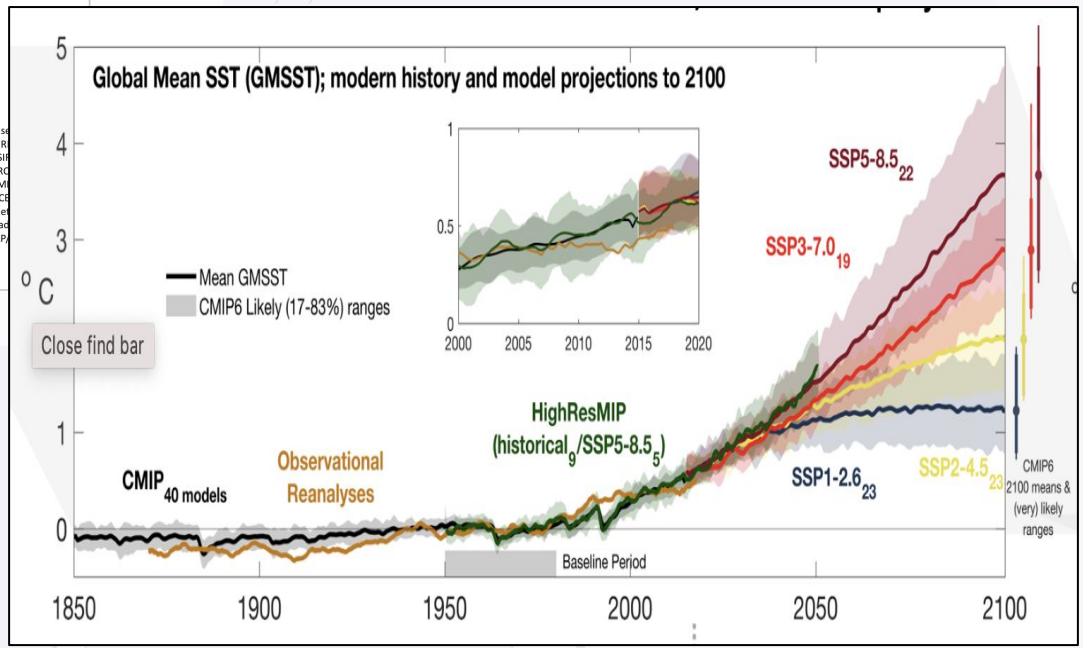
catalog with 1 dataset(s) from 3 asset(s):

Few lines of code later..

Data catalogs? intake-esm?



Cr. CIMES internship, Mackenzie Blanusa



Cr. IPCC and xMIP from Julius Busecke

A notebook example that “uses” the catalog

https://nbviewer.org/github/wrongkindofdoctor/MDTF-diagnostics/blob/refactor_pp/diagnostics/example_multicase/example_multirun_demo.ipynb

(GitHub reference)

https://nbviewer.org/github/aradhakrishnanGFDL/canopy-cats/blob/main/notebooks/om_example.ipynb

(GitHub reference)
(catalog example here)

Please contribute notebook examples that use GFDL generated catalogs and intake-esm [Issue page]

(Homework) Binder link Once the link loads, go to notebooks and run the demo-search-explore cell by cell.
GitHub reference)

More examples from the community:
https://github.com/aradhakrishnanGFDL/gfdl-aws-analysis
https://easy.gems.dkrz.de/Processing/Intake/index.html
https://gallery.pangeo.io/repos/pangeo-gallery/cmip6/intake_ESM_example.html
https://github.com/MackenzieBlanusa/OHC_CMIP6

Catalog Builder

- A “python community package ecosystem” that allows you to generate data catalogs compatible with intake-esm. Available as a Conda package (intakebuilder, scripts)

```
import intakebuilder
from scripts import gen_intake_gfdl
from intakebuilder import gfdlcrawler, CSVwriter, builderconfig, configparser
```

- Use it from a Jupyter notebook, a Python script, or from the command-line.
- **History:** Student intern research project - analysis using CMIP6 models.
- Catalog Builder GitHub repository
  - Automated build and testing (need more!)
  - Automated documentation
- Cite our work

Radhakrishnan, A., Brown, C., Monge, R., Chang, B., Blanton, C., & Sentman, L. (2024). Catalog Builder for data discovery and analysis at GFDL (Version v03.2024) [Computer software]. https://doi.org/10.5281/zenodo.10787602

Documentation

https://aradhakrishnangfdl.github.io/CatalogBuilder/generation.html

Quickstart

1. Activate conda environment

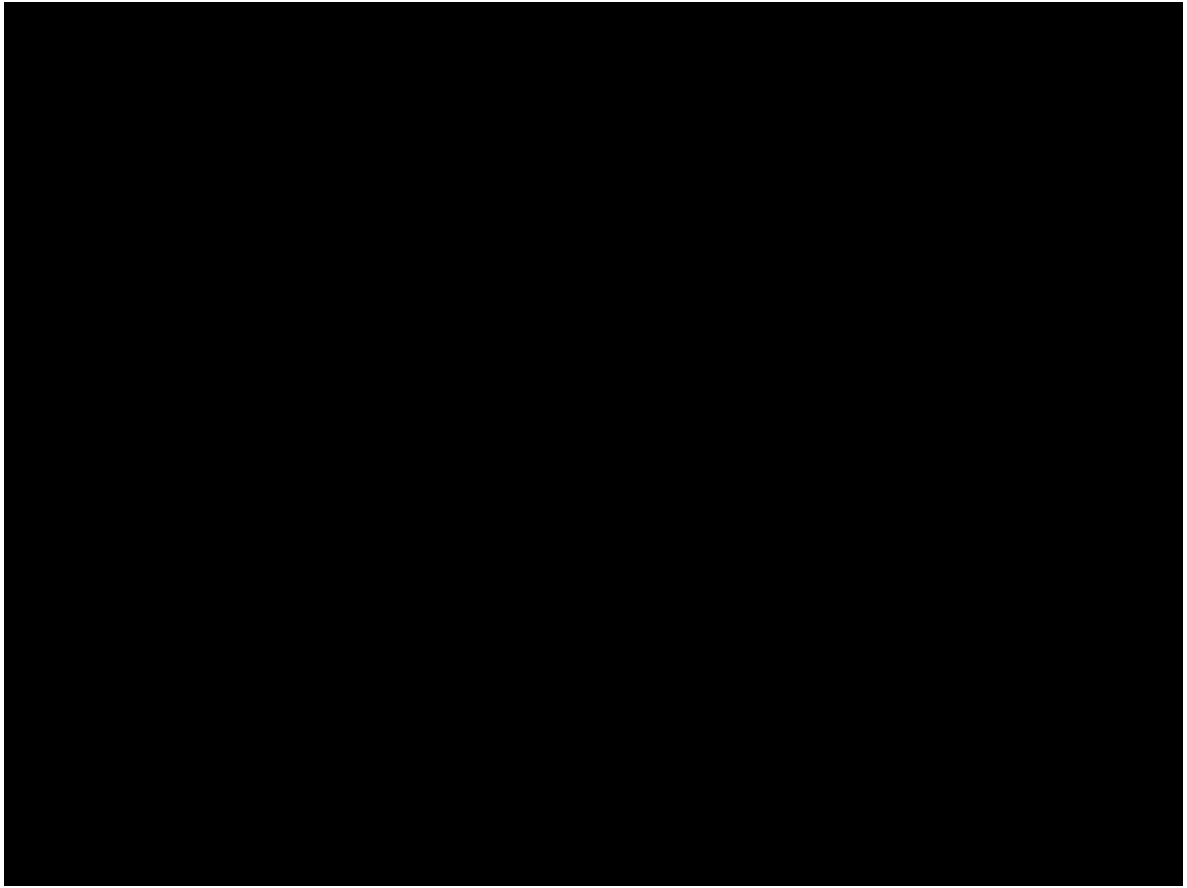
2. Add environment site package dir to PATH
   Ex. setenv PATH ${PATH}:${CONDA_PREFIX}/lib/python3.1/site-packages/scripts/

3. Call the script
   a. Must use this syntax: gen_intake_gfdl <input_path> <output_path>
   Ex. gen_intake_gfdl.py ~/path_to_data ~/output

This would create an output.CSV and an output.JSON in the user's home directory.

Overwriting and appending operations enabled through flags

Conda Package Demonstration



Configuration

Catalog headers (column names) are set with the **HEADER LIST** variable

The **OUTPUT PATH TEMPLATE** variable controls the expected directory structure of input data

Both can be configured by editing [intakebuilder/builderconfig.py]

Headerlist

activity_id	institution_id	source_id	experiment_id	frequency	modeling_realm	table_id	member_id
dev			c96L65_am5f3b	3hr	atmos_cmip		n/a
dev			c96L65_am5f3b	3hr	atmos_cmip		n/a
dev			c96L65_am5f3b	3hr	atmos_cmip		n/a

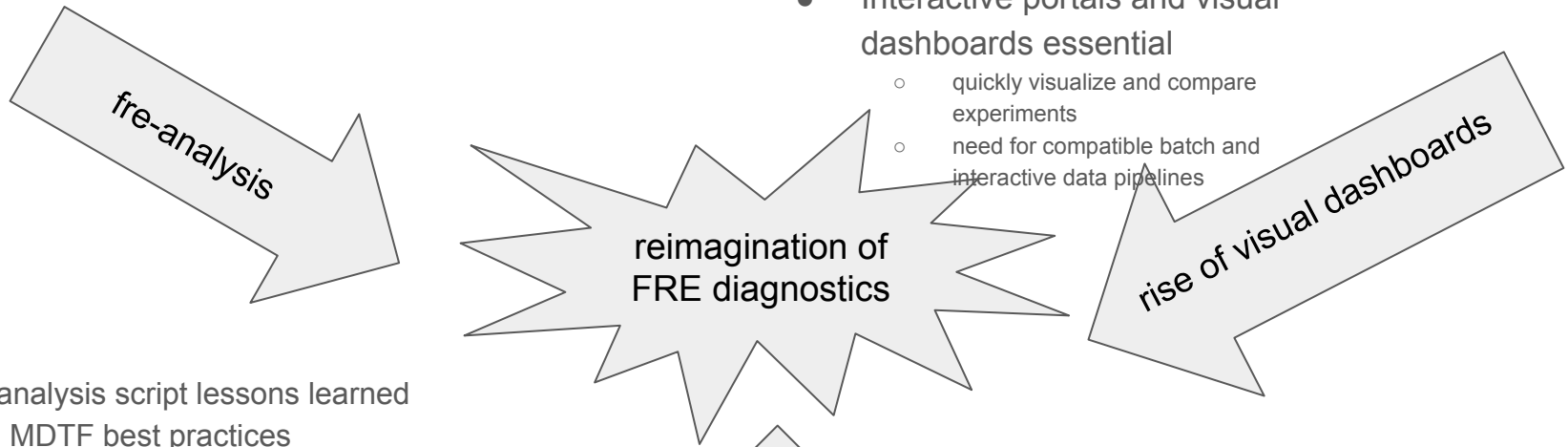
We set N/A for values that do not match up with what is expected

Future

https://github.com/aradhakrishnanGFDL/CatalogBuilder/issues

Repository will be under the NOAA-GFDL organization in the future

FRE opportunities

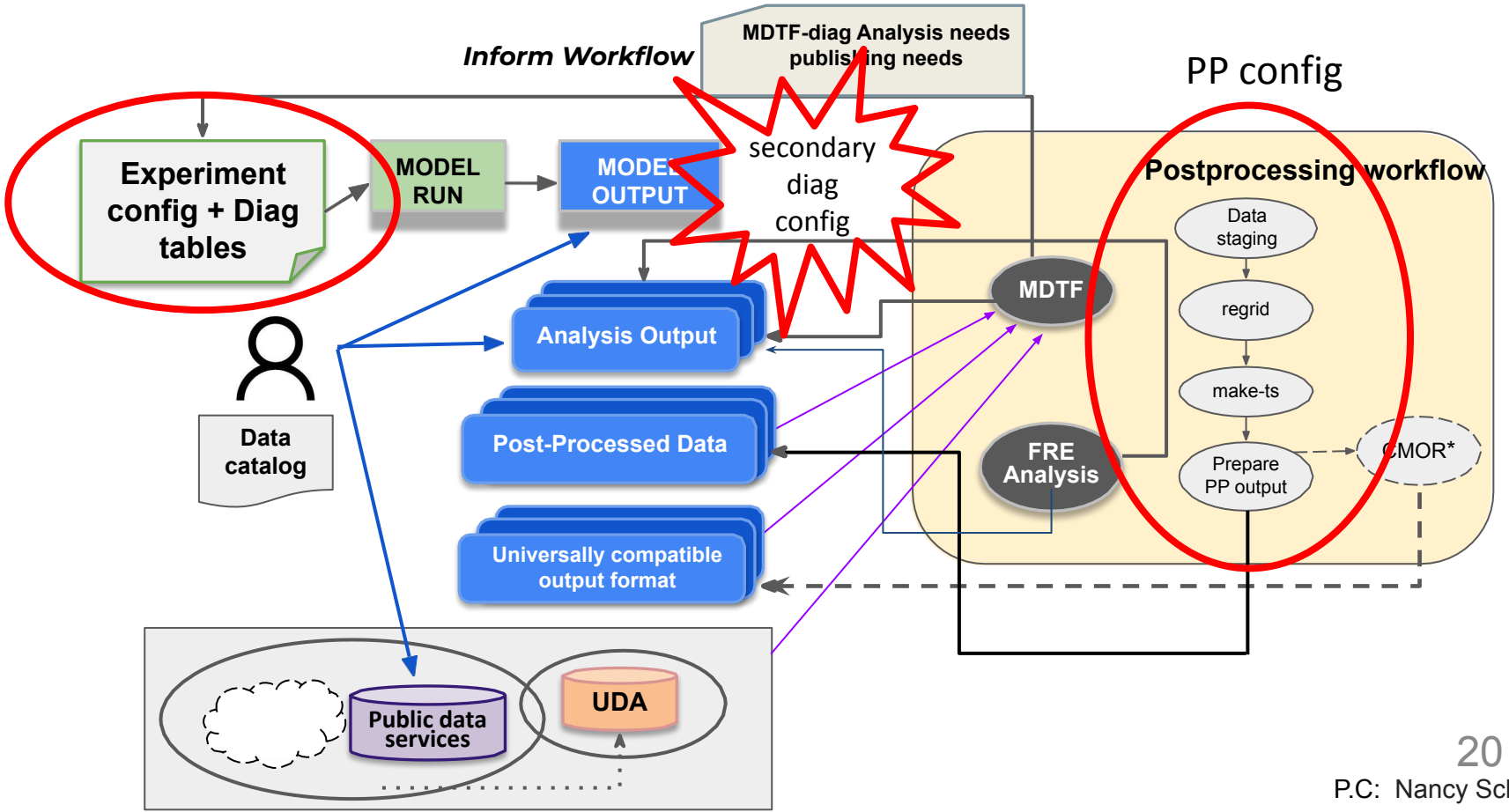


- fre-analysis script lessons learned and MDTF best practices
  - generation of fre-analysis scripts on life support
  - emerging analysis script best practices for portability, interoperability, and maintainability

- Interactive portals and visual dashboards essential
  - quickly visualize and compare experiments
  - need for compatible batch and interactive data pipelines

- CMIP and CMIP-like metadata standards
  - “CMIP-like” metadata is the de-facto climate data standard
  - separate data provider (gfdl or cmip-style) and data user concerns

New postprocessing workflow capabilities under development



P.C: Nancy Schultz

Envisioned path forward

- Leverage MDTF for GFDL
  - Encourage POD development and MDTF usage in FRE
  - Collaborate with MDTF framework developers for FRE integration features
- Mediate all workflow (FRE Canopy) data connection pathways through data catalogs
  - PP datasets available through vocabulary, not filesystem
  - Secondary diagnostics to be exposed through same standard API
  - Curated observation (UDA) datasets
  - Workflow verification of data pathway consistency before frerun
- Adopt CMIP-like metadata vocabulary
  - Not a single, one-time specification, but a living community standard
  - Remove GFDL/CMIP distinctions (from workflow configuration and analysis scripts)

- FRE analysis subgroup: bottom-up group of interested GFDL users for collaboration, brainstorming, consensus-building
  Every other Wednesday at 11am in 129 (tomorrow only in 317)

FRE and MDTF

- FRE and MDTF
  - Use MDTF for FRE Canopy analysis script batch launching
  - Facilitate MDTF usage for FRE users (e.g. easy opt-in switch)
  - Socialize MDTF path for FRE analysis developers: take advantage of MDTF without “full POD membership”
  - (This proposed FRE plugin capability would be partway/ compatible development towards “full POD membership”)
- Co-develop information pathways with FRE
  - i.e. frerun check for data pathway consistency
  - connection to existing MDTF capability (settings.json)

MDTF established community development

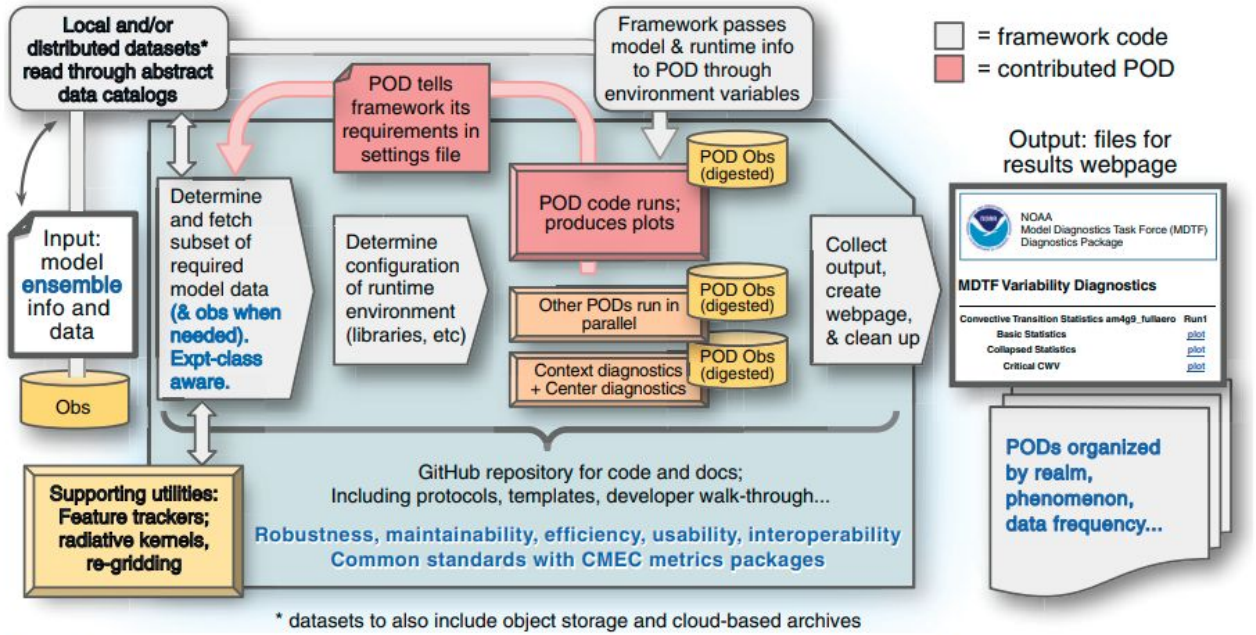


Fig. 1. Diagram of the MDTF framework evolving under the current phase of development. The framework manages a set of process-oriented diagnostics modules (PODs) contributed by a variety of diagnostic development teams. Coordinated standards facilitate the inclusion/exchange of metrics and diagnostics with other parts of the U.S. diagnostics community.

Neelin, J. D., and Coauthors, 2023: Process-Oriented Diagnostics: Principles, Practice, Community Development, and Common Standards. *Bull. Amer. Meteor. Soc.*, **104**, E1452–E1468, https://doi.org/10.1175/BAMS-D-21-0268.1.

PP datasets available through vocabulary, not filesystem

	Traditional filesystem access	intake-esm access
Locate collection	cd /archive/path/to/pp	col = intake.load('/archive/path/to/catalog/json')
Search collection & locate desired items	ls ocean/daily/5yr | grep sst	cat = col.search(expname='myexp', comp='ocean', freq='daily', var_id='sst)'

Pass files to python	python script.py --file /path/to/file --var sst
	# within python script, load file and variable
	ds = xr.open_dataset
		dset_dict = cat.to_dataset_dict()
		ds = dataset_dict['myexp.daily.ocean.sst']

Use arrays	ds	ds

Secondary diagnostics to be exposed through same standard API

	Traditional filesystem access	intake-esm access
Locate collection	cd /archive/path/to/pp	col = intake.load('/archive/path/to/catalog.json')
Search collection & locate desired items	ls atmos_refined/daily/5yr | grep uu inflexible search	cat = col.search(experiment_id='myexp', modeling_**realm='atmos'**, frequency='daily', variable_id='uu')

Pass files to python	python script.py --file /path/to/file --var sst
	# within python script, load file and variable
	ds = xr.open_dataset
		dset_dict = cat.to_dataset_dict()
		ds = dataset_dict['myexp.daily.ocean.uu]

Use arrays	ds	ds

flexible search

Curated observation (UDA) datasets

	Traditional filesystem access	intake-esm access
Locate collection	cd /archive/uda/ERA5	col = intake.load('/archive/uda/catalog.json')
Search collection & locate desired items	ls Hourly_Data_On_Pressure_Levels/reanalysis/global/700hPa/6hr-timestep/annual_file-range/relative_humidity/	cat = col.search(uda='ERA5', modeling_realm='atmos', frequency='6hr', variable_id='rh')

Pass files to python	python script.py --file /path/to/file --var sst
	# within python script, load file and variable
	ds = xr.open_dataset
		dset_dict = cat.to_dataset_dict()
		ds = dataset_dict['myexp.daily.ocean.uu]

Use arrays	ds	ds

CMIP-like model output metadata vocabulary

- Some convention needed to get work done, avoid chaos
  - Is CMIP-X vocabulary less fragile than legacy Bronx frepp vocabulary? No (but it's still worthwhile)
- CMIP-like defined
  - Use CMIP-X vocabulary where possible
  - Seek at least GFDL consensus for other vocabulary not covered by CMIP-X; e.g. grid, dimensions, cell methods
- Add vocabulary metadata instead of rewriting
  - Key-value information superior to directories for cataloging needs
  - e.g. support alternate variable names (GFDL/CMIP data friction)

If interested:

  - FRE analysis subgroup: bottom-up group of interested GFDL users for collaboration, brainstorming, consensus-building
  - Every other Wednesday at 11am in 129

Envisioned path forward

- Leverage MDTF for GFDL
  - Encourage POD development and MDTF usage in FRE
  - Collaborate with MDTF framework developers for FRE integration features
- Mediate all workflow (FRE Canopy) data connection pathways through data catalogs
  - PP datasets available through vocabulary, not filesystem
  - Secondary diagnostics to be exposed through same standard API
  - Curated observation (UDA) datasets
  - Workflow verification of data pathway consistency before frerun
- Adopt CMIP-like metadata vocabulary
  - Not a single, one-time standard, but a living community standard
  - Remove GFDL/CMIP distinctions (from workflow configuration and analysis scripts)

- FRE analysis subgroup: bottom-up group of interested GFDL users for collaboration, brainstorming, consensus-building
  Every other Wednesday at 11am in 129 (tomorrow only in 317)

Supplementary slides

Xarray

- A high-level API for loading, transforming, and performing calculations on multi-dimensional arrays.
- Built leveraging NumPy and pandas API
- Code it like you say it (Easy to get started!)
  - E.g. `ds.sel(time='2000-01')`
    `ds['thetao'].sel(z_l=2.5).mean(dim='time')`
- Incentive, motivation to adhere to CF conventions.
  - Leverages the use of CF metadata Conventions
- Simple gateway to exploring different data formats and input sources.
  - E.g. NetCDF, OPeNDAP, Google cloud data store, Zarr,……
- xarray's data structures can be backed by dask

Gentle learning curve

xarray: originally developed by S. Hoyer.

https://github.com/pydata/xarray