IWSG 2022

GeoEDF: A Framework for Designing and Executing Reproducible Geospatial Research Workflows in Science Gateways

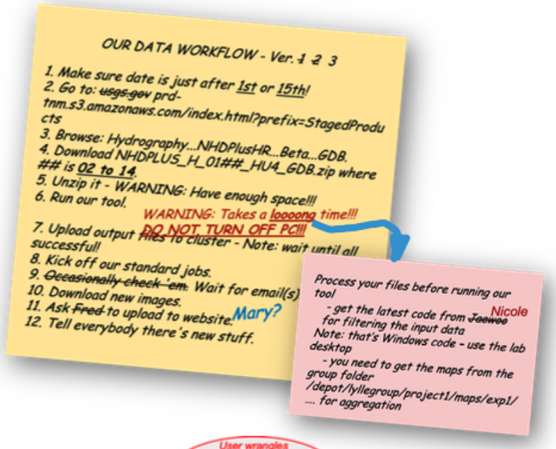
Rajesh Kalyanam, Jungha Woo, Lan Zhao, Carol X. Song, and Jack Smith*
Rosen Center for Advanced Computing, Purdue University
*Marshall University

PURDUE UNIVERSITY

June 15-17, 2022, Trento, Italy

GeoEDF Vision

Researchers spend up to 80% of their time "wrangling data"



Remote data repos, smart devices, streaming data

GeoEDF Geospatial Data Framework

- Reusable Data Connectors
- Pluggable Data Processors
- Integrated Active Learning
- Resource Interoperability Interfaces

Geospatial Data Pipeline Composer (GUI & API)

Cyberinfrastructure (Campus, XSEDE, HUBzero, Geospatial Tools, storage, Solr, ...)

Make Science FAIR

OUR DATA WORKFLOW - Final

1. Go to the science gateway
2. Define "my_workflow.yml" (or use tool GUI if needed)
3. Ask GeoEDF to execute!
4. Data and workflow automatically published to science gateway

- Remote data directly usable in code, seamless workflow
- Complexity abstracted away
- Reusable data connectors, processors, and workflows
- Automatic provenance capture & data annotation => FAIR

GeoEDF Project

An Extensible Geospatial Data Framework Towards FAIR Science

To help data-driven sciences to be more
Findable, Accessible, Interoperable, Reusable

funded by NSF CSSI program award #: 1835822, Oct 2018 - Sep 2023

GeoEDF Components

Reusable Data Connectors

Implement various data access protocols, enable data acquisition from popular repositories

+

Reusable Data Processors

Implement domain agnostic & domain specific geospatial processing operations

+

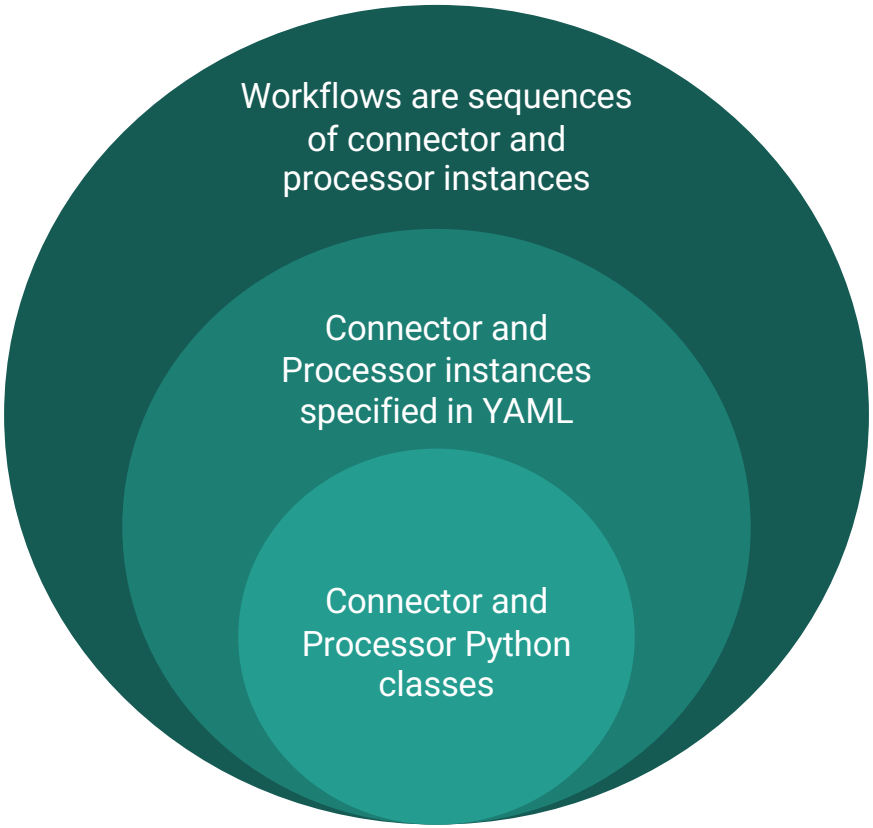
Plug-and-play Workflow Composer

Enable the composition of individual connectors & processors into complex workflows

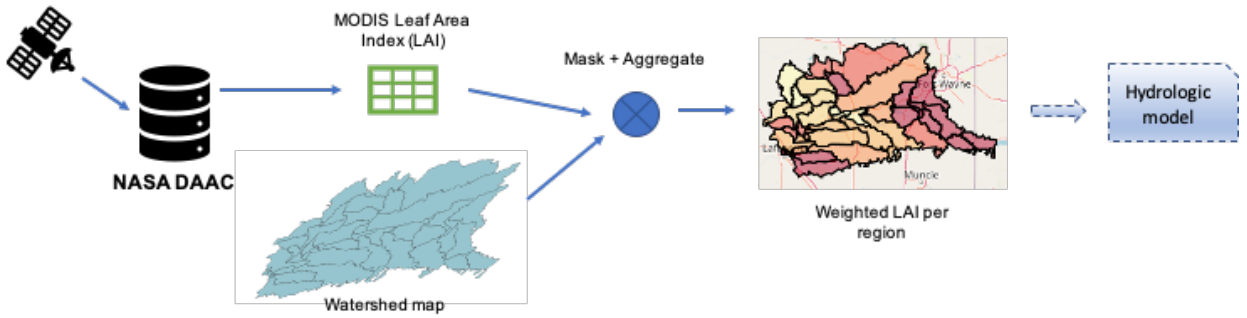
GeoEDF

Enable researchers to conceive of geospatial data driven workflows as a sequence of data acquisition and processing steps that can be carried out using pre-existing or user contributed connectors and processors

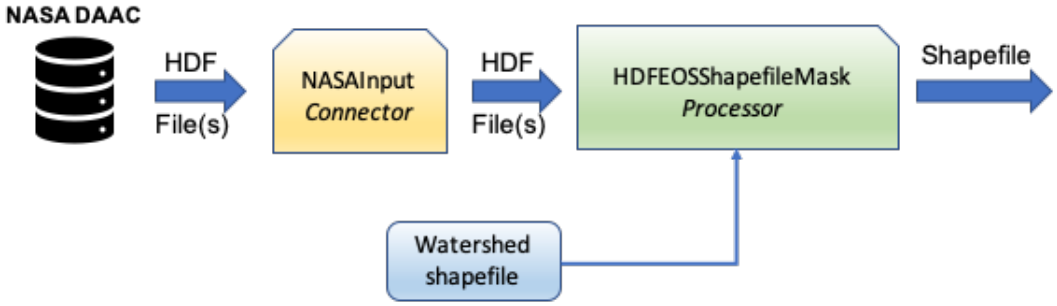
GeoEDF Workflow in a nutshell



Example Hydrologic Workflow



Apply GeoEDF principles



Corresponding GeoEDF Workflow (YAML)

Data connector

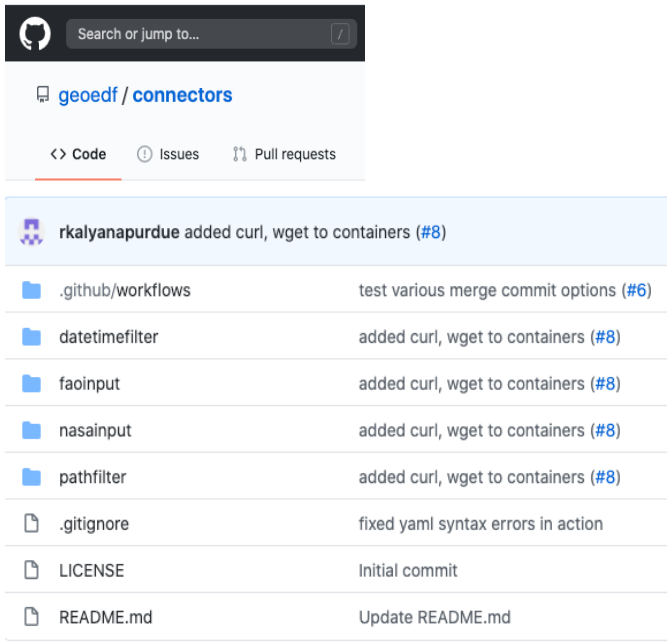
```
$1:
  Input:
    NASAInput:
        url: https://e4ftl01.cr.usgs.gov/MOTAMCD15A3H.006/%{file}
        user: rkalyana
        password:
  Filter:
    file:
      PathFilter:
        pattern: '%{dtstring}/MCD15A3H.*.h09v07*.hdf'
    dtstring:
      DateTimeFilter:
        pattern: '%Y.%m.%d'
        start: 07/16/2002
```

- Filters enable spatial and temporal filtering before data acquisition
- This improves workflow generality and efficiency

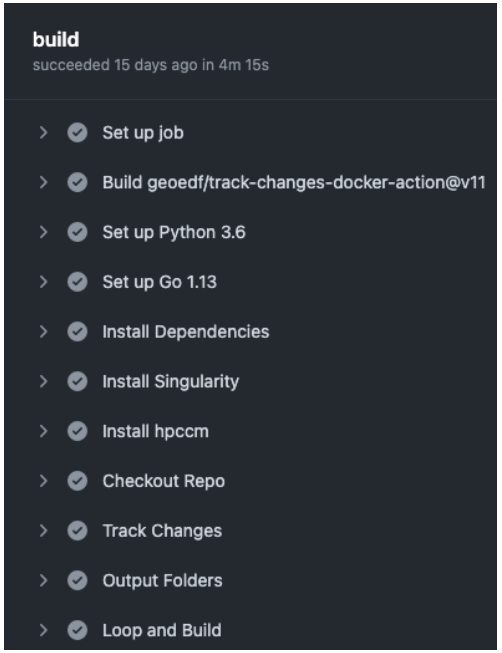
Data processor

```
$2:
    HDFEOSShapefileMask:
        hdffile: $1
        shapefile: /home/mygeohub/rkalyana/subs1_projected_171936.shp
        datasets: [Lai]
```

Connector, Processor Contribution Process



(1) Contribute connectors/processors via GitHub pull requests

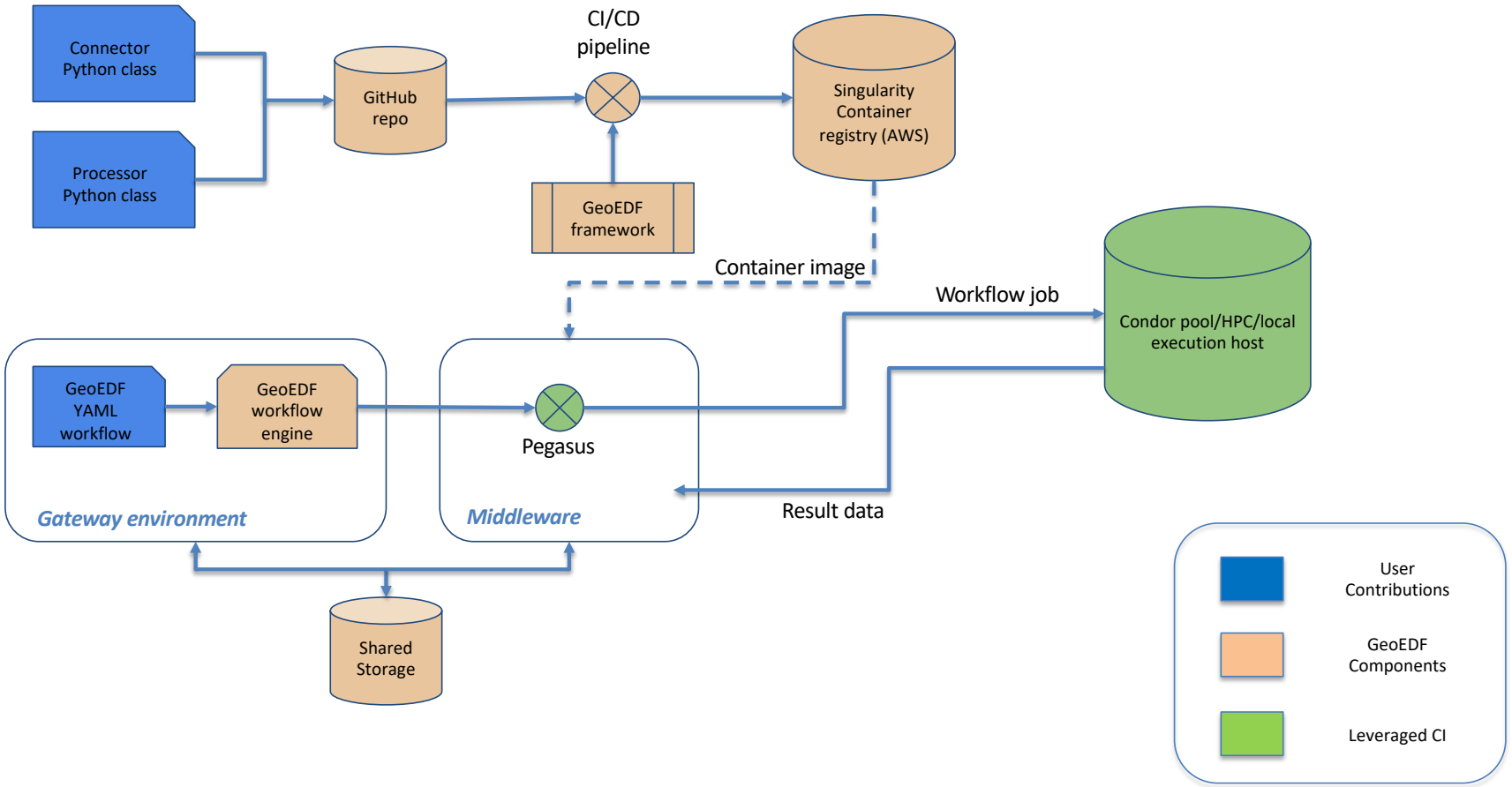


(2) GitHub action detects changes, builds Singularity container, pushes to registry server

```
def get_registry_containers(self):
    cli = get_client(quiet=True)

    conns = dict()
    query_res = cli.search("connectors")
    for (cont_uri,url) in query_res:
        cont_path = cont_uri.split(':')[0]
        plugin_name = cont_path.split('/')[1]
        if plugin_name not in conns:
            conns[plugin_name] = cont_uri

    procs = dict()
    query_res = cli.search("processors")
    for (cont_uri,url) in query_res:
        cont_path = cont_uri.split(':')[0]
        plugin_name = cont_path.split('/')[1]
        if plugin_name not in procs:
            procs[plugin_name] = cont_uri

    return (conns,procs)
```

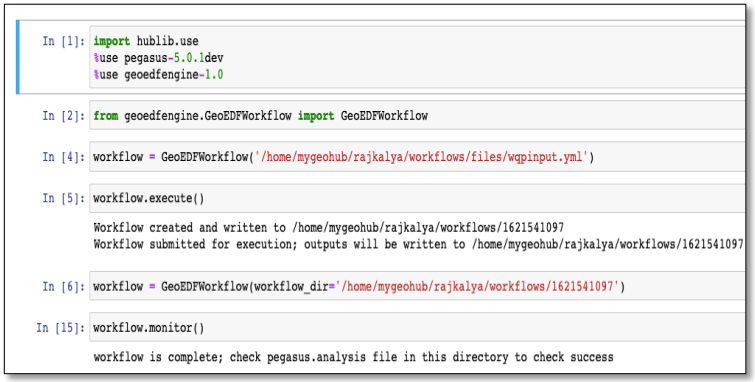
(3) Workflow engine queries registry for list of connector, processor containers

Gateway Integration



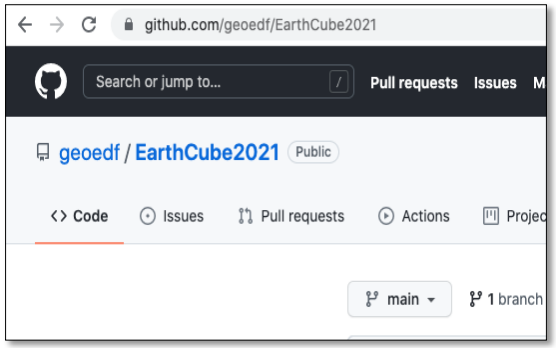
Deployment Solutions

MyGeoHub



- Publicly available gateway
- Deployed in Jupyter notebook environment as a Python library
- Job submission to Purdue's Halstead cluster

docker



- Self-contained Docker container
- Can use to build and test new connectors, processors
- Run on your own machine

jupyterhub

- Standalone deployment, in the works…
- CILogon authentication
- Workflow execution in local minicondor

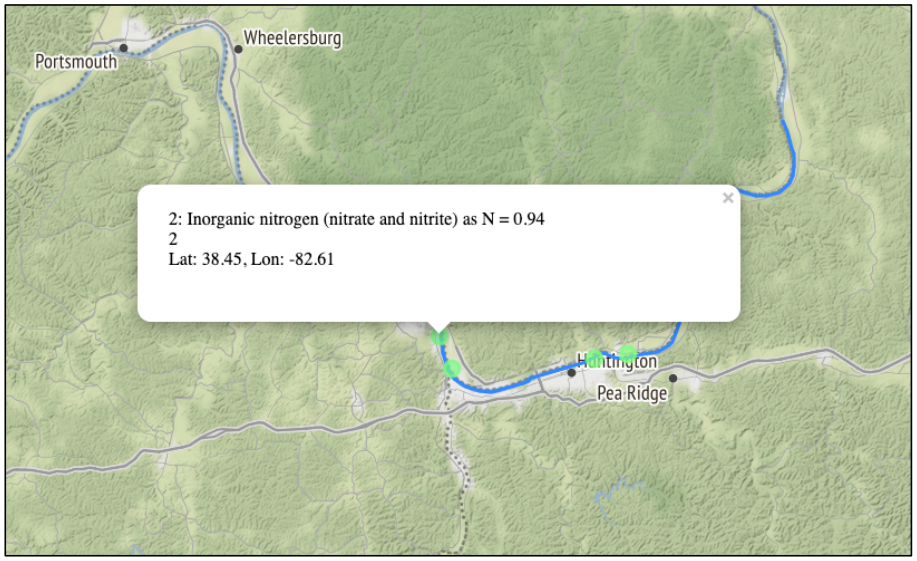
GeoEDF Demo

GEOEDF WORKFLOW EXECUTION IN THE MYGEOHUB GATEWAY

GeoEDF Applications – Water Quality

Synthesize hydrologic and water quality data from various federal agencies (USGS, EPA, etc.) for EPSCoR states for ease of visualization and analysis

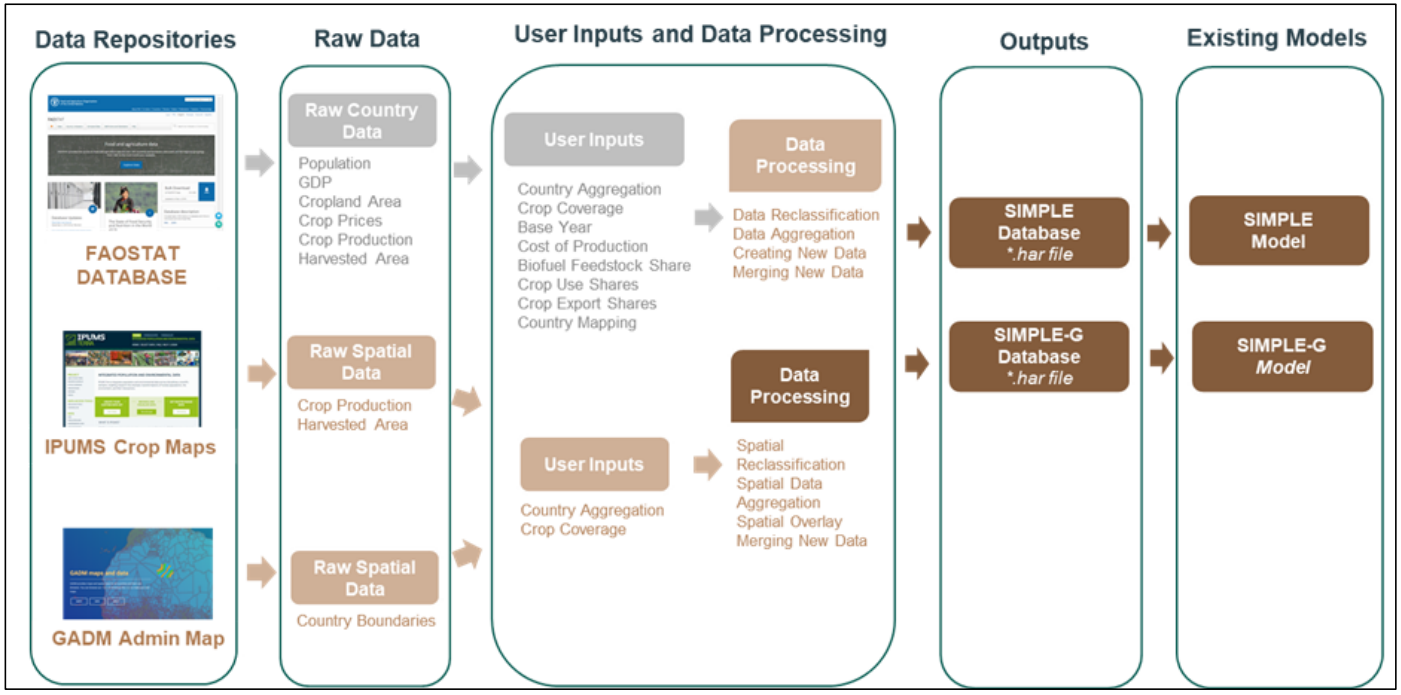
- Workflow produces an interactive map combining water quality data from WQP and stream reach data for a given monitoring station



GeoEDF Applications – Agricultural Economics

Acquire and pre-process the necessary socio-economic, agricultural, and climate data for analyzing global-to-local food security and sustainability

- Workflow acquires diverse U.N. FAOSTAT datasets, aggregates it for the study region, and converts from custom "HAR" format into widely-used csv

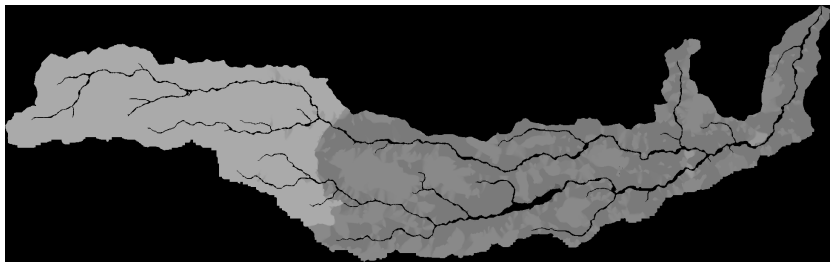


GeoEDF Applications – InVEST

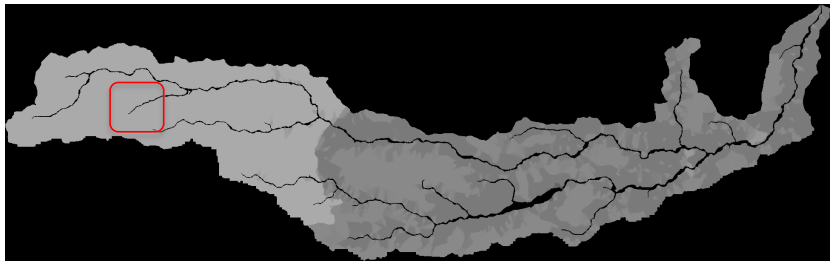
Family of tools (models) for quantifying importance of natural capital

- Workflow wraps the InVEST NDR (nutrient delivery ratio) model to enable efficient parameter sweeps via HPC execution

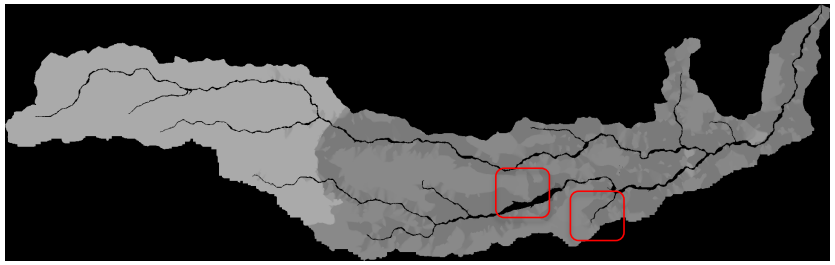
threshold_flow_accumulation = 1000



threshold_flow_accumulation = 1500



threshold_flow_accumulation = 2000



Summary

- Research reproducibility via declarative workflow conceptualization
- Broad applicability across domains that have workflows with a mix of data acquisition and processing steps
- Variety of integration options with CI and gateway platforms
- Ability to leverage various compute resources (local machine, Condor pool, HPC)

Thank You

- GeoEDF GitHub Repository: https://github.com/geoedf
- GeoEDF Documentation: https://geoedf.readthedocs.io
- Publication: https://dl.acm.org/doi/10.1145/3311790.3396631

Email:
Carol Song (cxsong@purdue.edu)