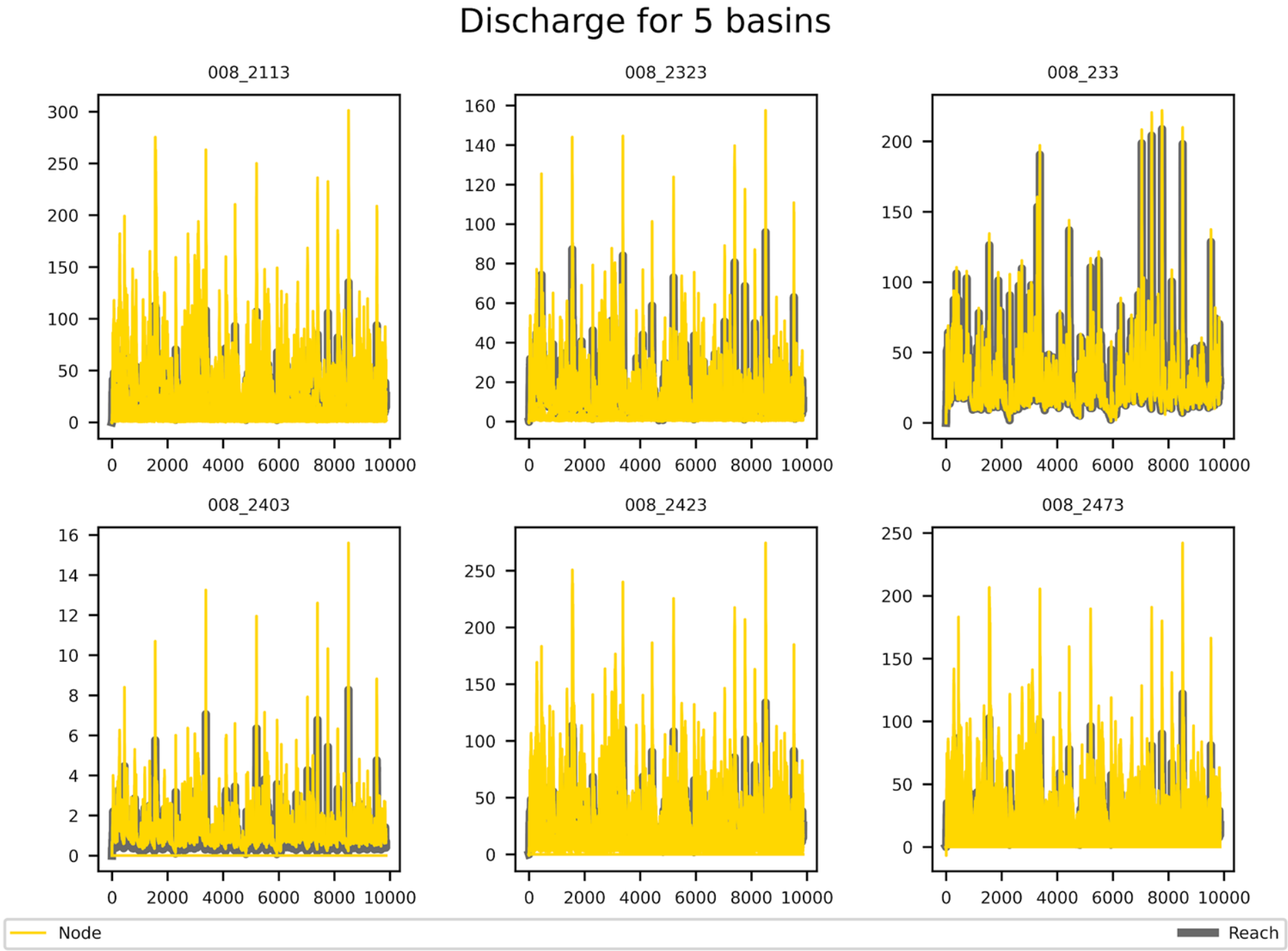
SWOT, SWORD of Science (SoS), and Confluence

SWOT DAWG 03/16/2021

Model data

Euro LISFLOOD data from Paul Bates and Jeison Sosa

After data cleaning:
Total reaches: 2,085
Total nodes: 165,933
Total time: 9,362



SWOT-like Data

We generated a NetCDF to represent SWOT observation data variables or attributes for each reach:

- d_x_area
- slope2
- Width
- Wse
- Generated on the node AND reach level

Each NetCDF variable closely matches the SWOT product description for the SWOT observation data that we will receive from PO.DAAC/JPL

SWOT NetCDF

Dimensions

Name	Description	Value (if applicable)
nt	Time steps	9362
nx	Nodes	Varies by reach

Groups

reach
node

Variables

reach/d_x_area		
	dimensions	nt
	type	float
	long_name	Change in cross-sectional area
	units	m^2
	valid_min	-10000000
	valid_max	10000000
	fill_value	-9999

node/d_x_area		
	dimensions	nx by nt
	type	float
	long_name	Change in cross-sectional area
	units	m^2
	valid_min	-10000000
	valid_max	10000000
	fill_value	-9999

SoS Data

- We also generated a NetCDF to represent SoS data for each reach.
- SoS data currently consists of priors from geoBAM.
  - This mostly includes scalar data on the reach level
  - River type priors are stored as vectors on the node level
- The SoS will continue to evolve and grow over time.

SoS data (geoBAM priors)		
qhat	lowerbound_A0	upperbound_A0
lowerbound_logn	upperbound_logn	lowerbound_b
upperbound_b	lowerbound_logWb	upperbound_logWb
lowerbound_logDb	upperbound_logDb	lowerbound_logr
upperbound_logr	logA0_hat	logn_hat
b_hat	logWb_hat	logDb_hat
logr_hat	logA0_sd	logn_sd
b_sd	logWb_sd	logDb_sd
logr_sd	lowerbound_logQ	upperbound_logQ
lowerbound_logWc	upperbound_logWc	lowerbound_logQc
upperbound_logQc	logWc_hat	logQc_hat
logQ_sd	logWc_sd	logQc_sd
Werr_sd	Serr_sd	dAerr_sd
sigma_man	sigma_amhg	

SWORD of Science NetCDF

Global Attributes

Name	Description	Value (if applicable)
valid	Indicated if the reach is valid	1 for valid or 0 for invalid

Dimensions

Name	Description	Value (if applicable)
nx	Nodes	Varies by reach

Groups

reach
node
inversion_set

Variables

reach/Qhat		
	dimensions	scalar
	type	float
	long_name	Mean_Q
	units	m^3/s
	fill_value	-9999

node/river_type		
	dimensions	nx
	type	float
	long_name	Brinkerhoff_class_number
	units	NA
	fill_value	-9999

SoS: Building

- Data can be added to the SoS as requirements are refined for each algorithm

- To add to the SoS:
  - Provide either reach_id or node_id from SWORD so that the data can be easily mapped to SWORD and topology data
  - Provide the dimensions of your data for each NetCDF variable
  - Provide associated metadata descriptions for each NetCDF variable trying to match what we have currently and add as necessary
  - Provide the group (e.g., “node”, “reach”) where your data should be stored

SoS: Access

- The SoS is currently stored in OneDrive until the Ernesto/Kevin/Sophie/Colin group weighs in
- Please email ntebaldi@umass.edu for access to the compressed file

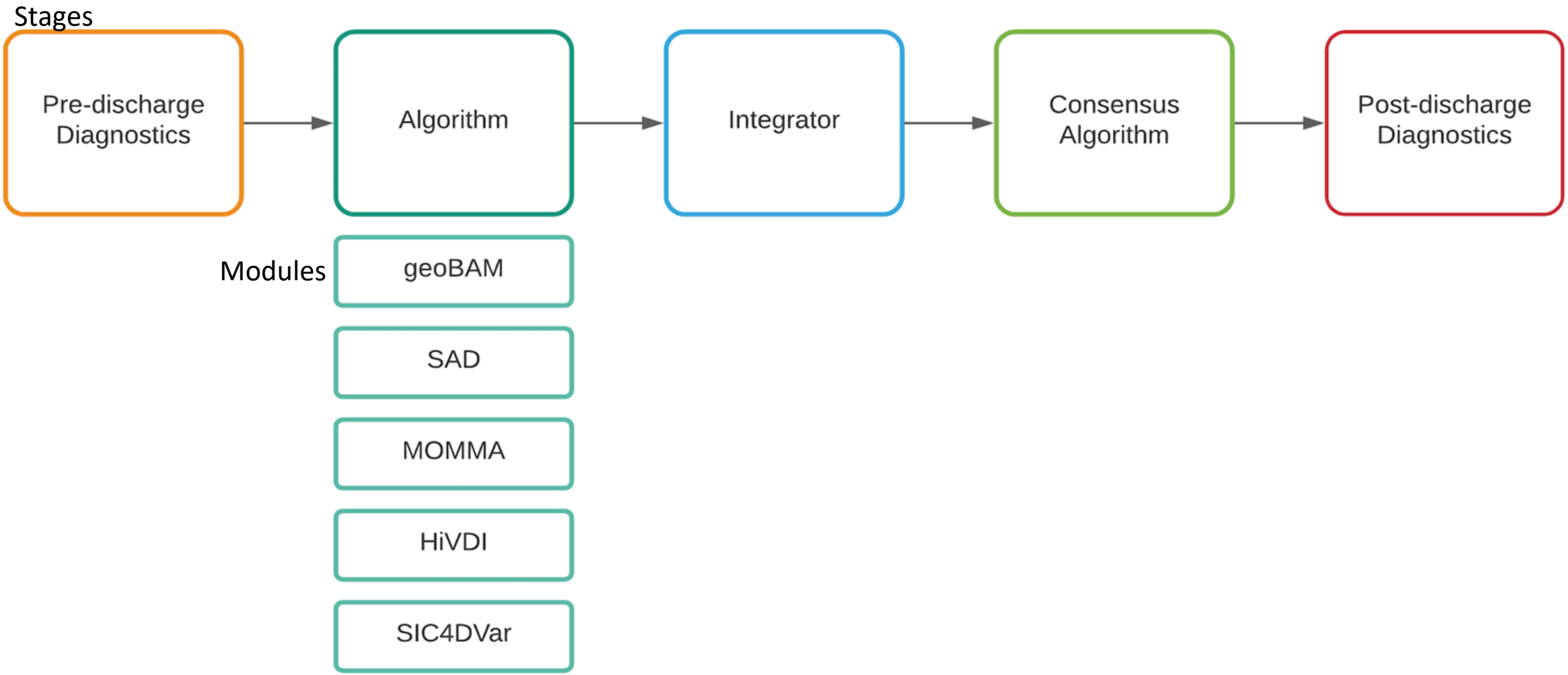
Topology data

Columns

Column Name	Description
index	index value for node
lon	longitude value for node
lat	latitude value for node
link	reach identifier
dslink	downstream reach identifier

Confluence

Confluence is a Python program that builds and runs containers; it contains stages and modules where each module will be a container

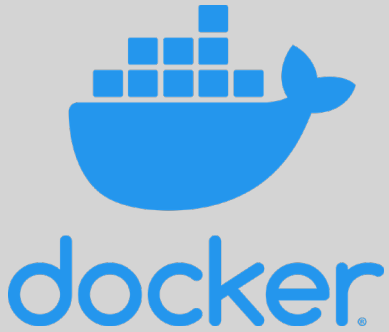


Confluence: Structure

Confluence is flexible and modular

You can run only what you need as Docker or Singularity containers or from within Confluence

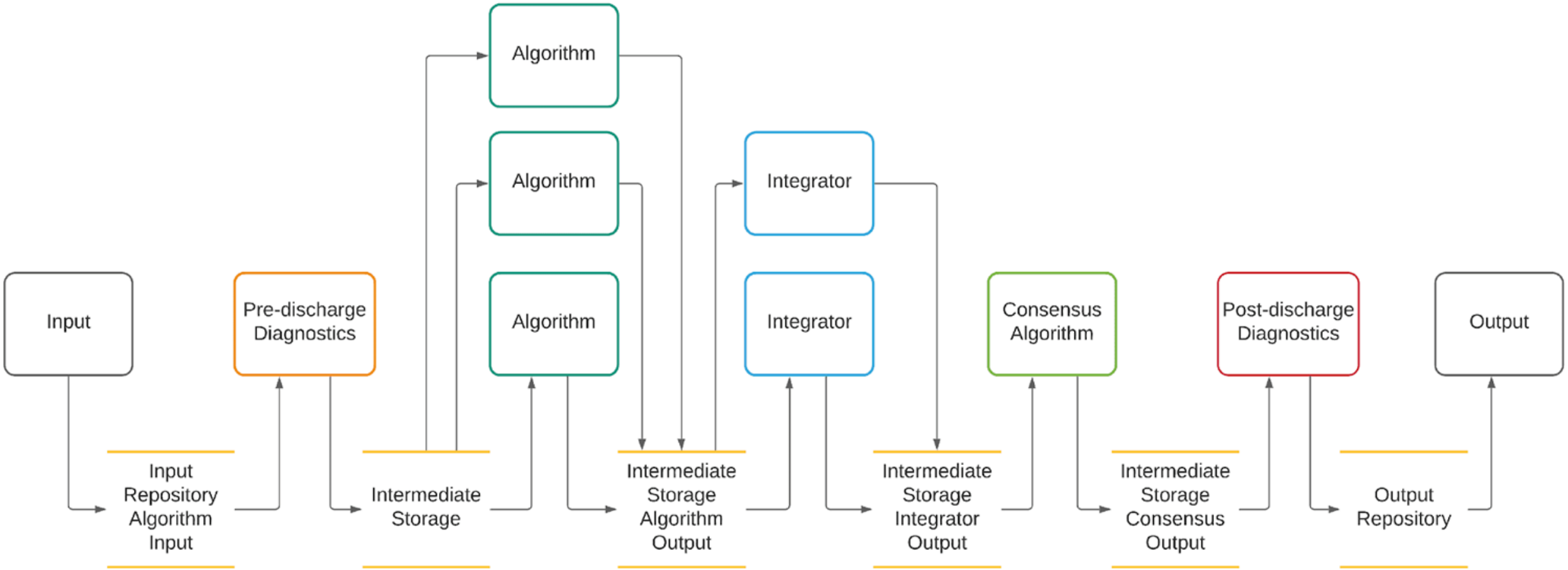
```
config = {
    "algorithms" : ["geobam"],
    "working_path" : ""
}
```



Confluence: Data

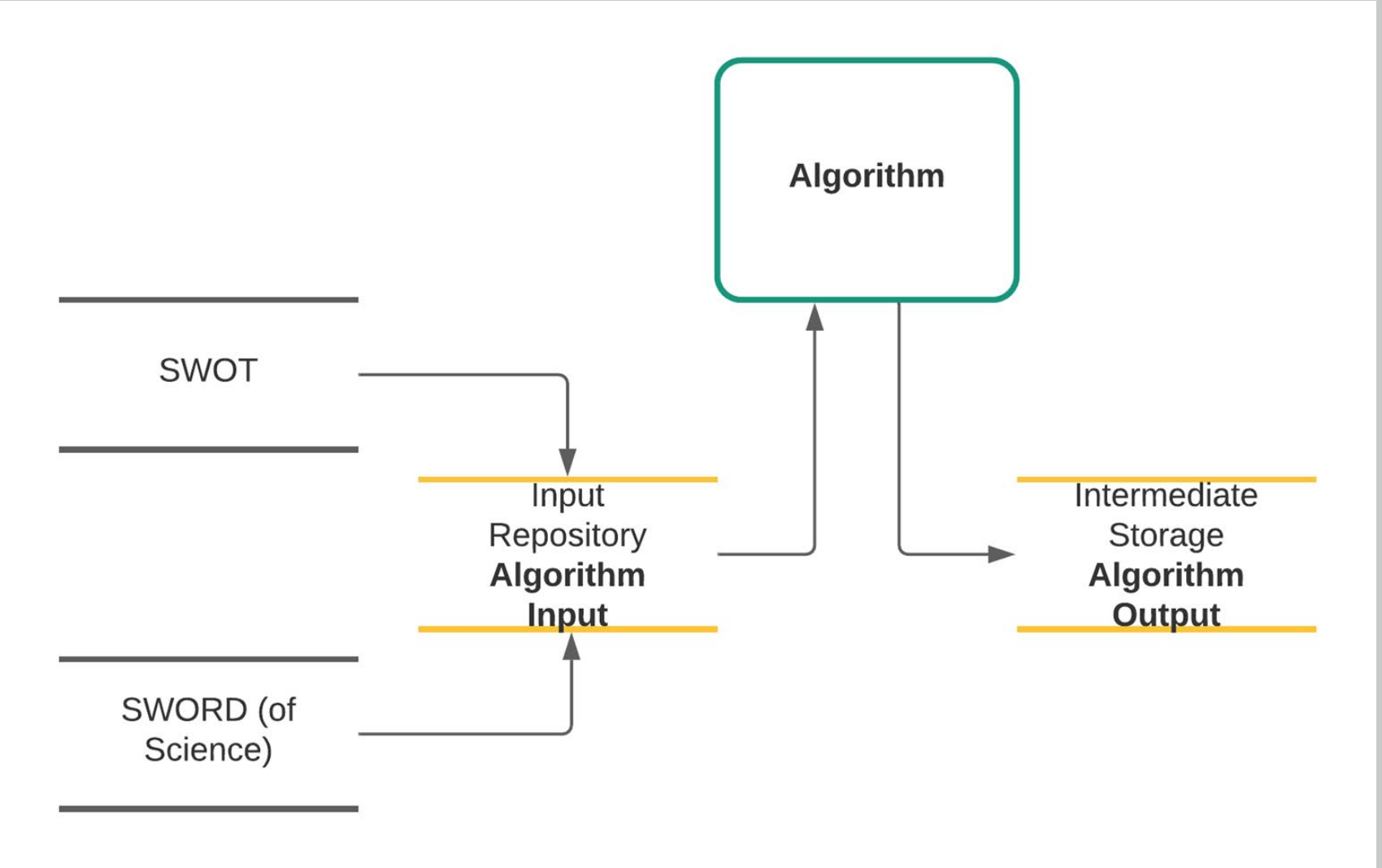
- Data will be taken from SWOT, SWORD, and the SoS
- It will be stored in intermediate forms (currently NetCDF) and fed to each stage
  - Each module in each stage will need to accept input and produce output
- A0 and n will be sent to SWORD but other output data can be stored in the SoS

Confluence: Data Flow

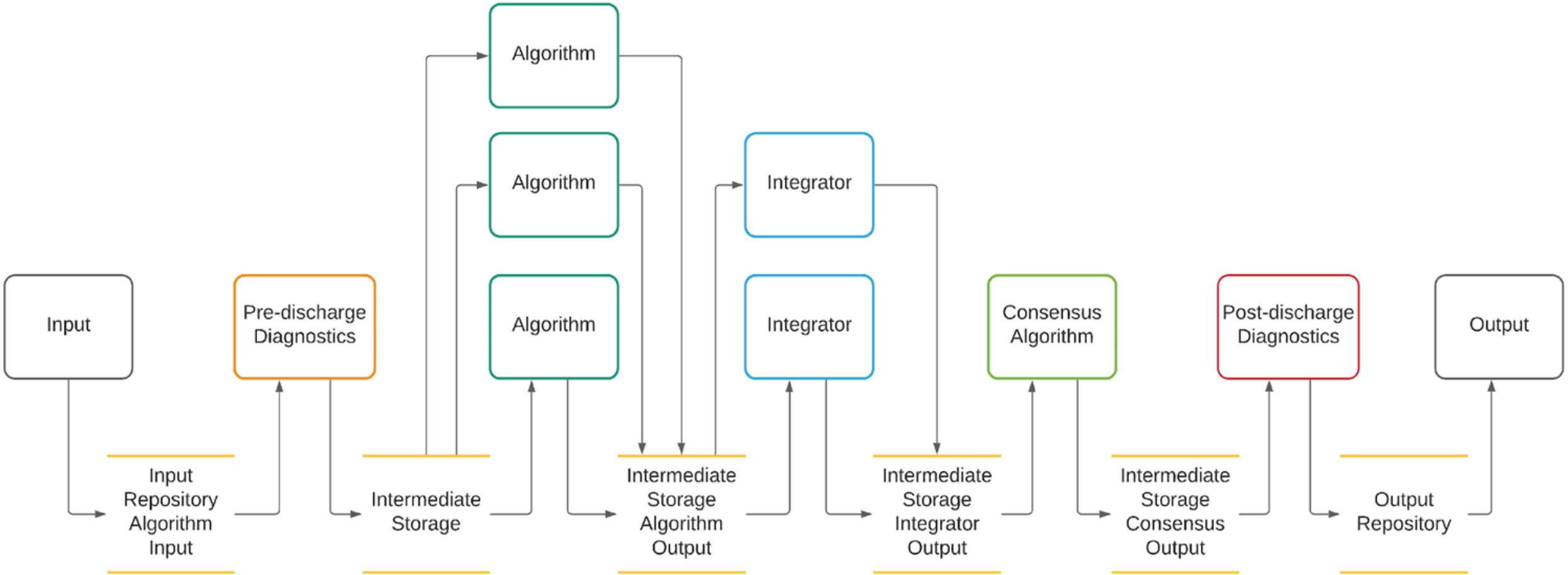


Confluence: Current Focus

- Development is currently focused on the ingestion of algorithms
- I am working with algorithm developers to understand input and output so containerize code to accept input and produce output in a consistent manner



Questions and Feedback



contact: ntebaldi@umass.edu