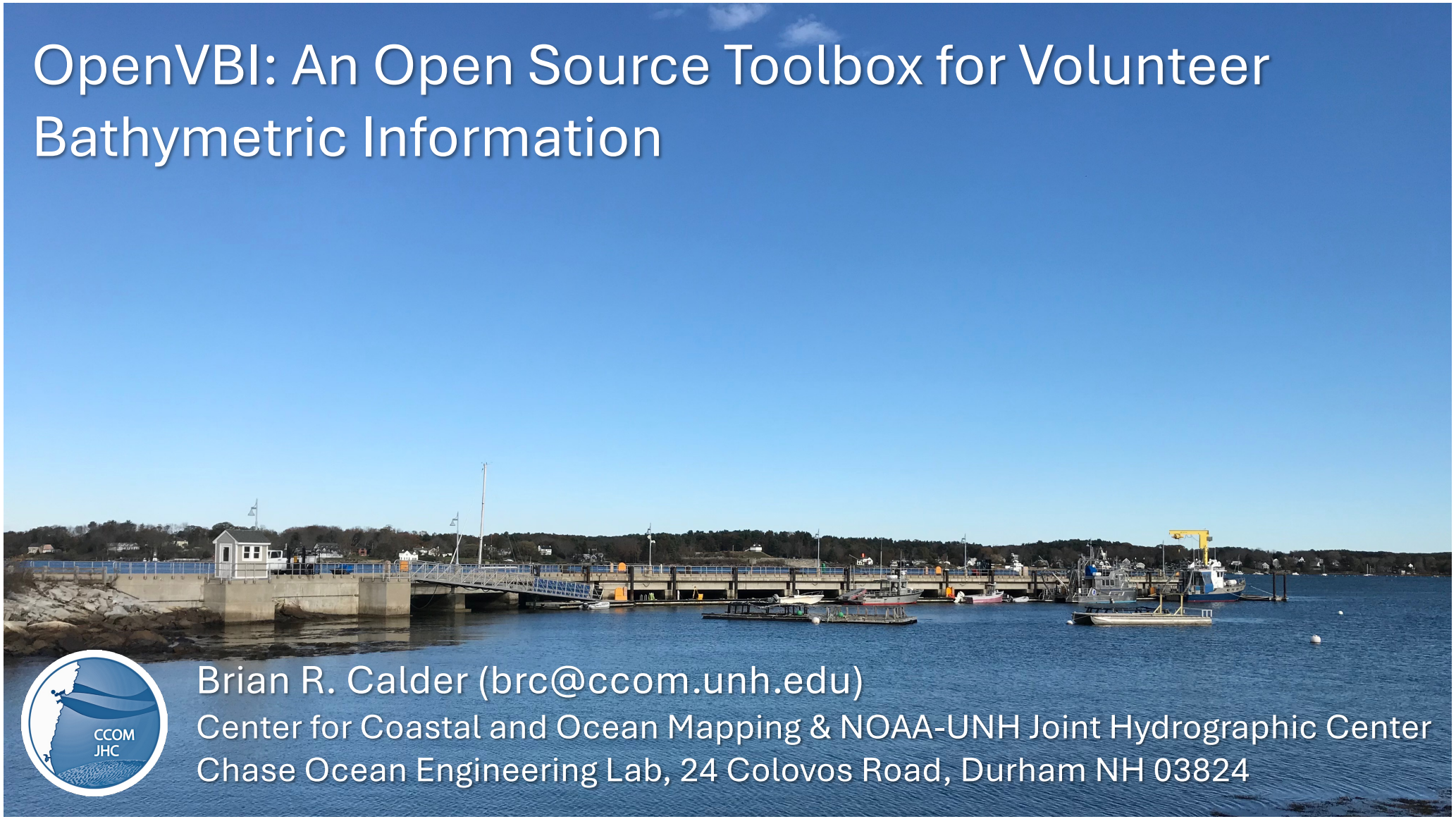
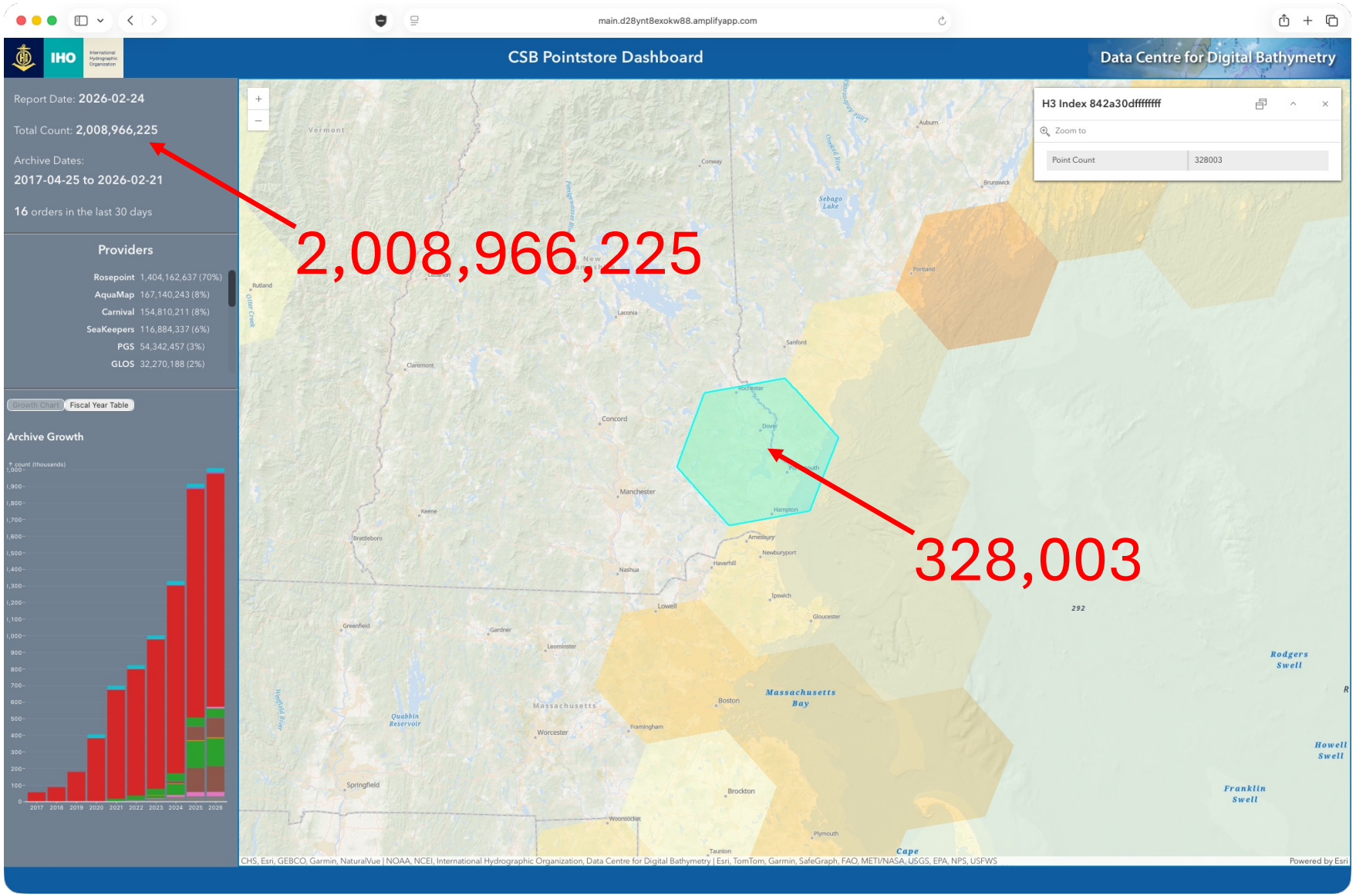
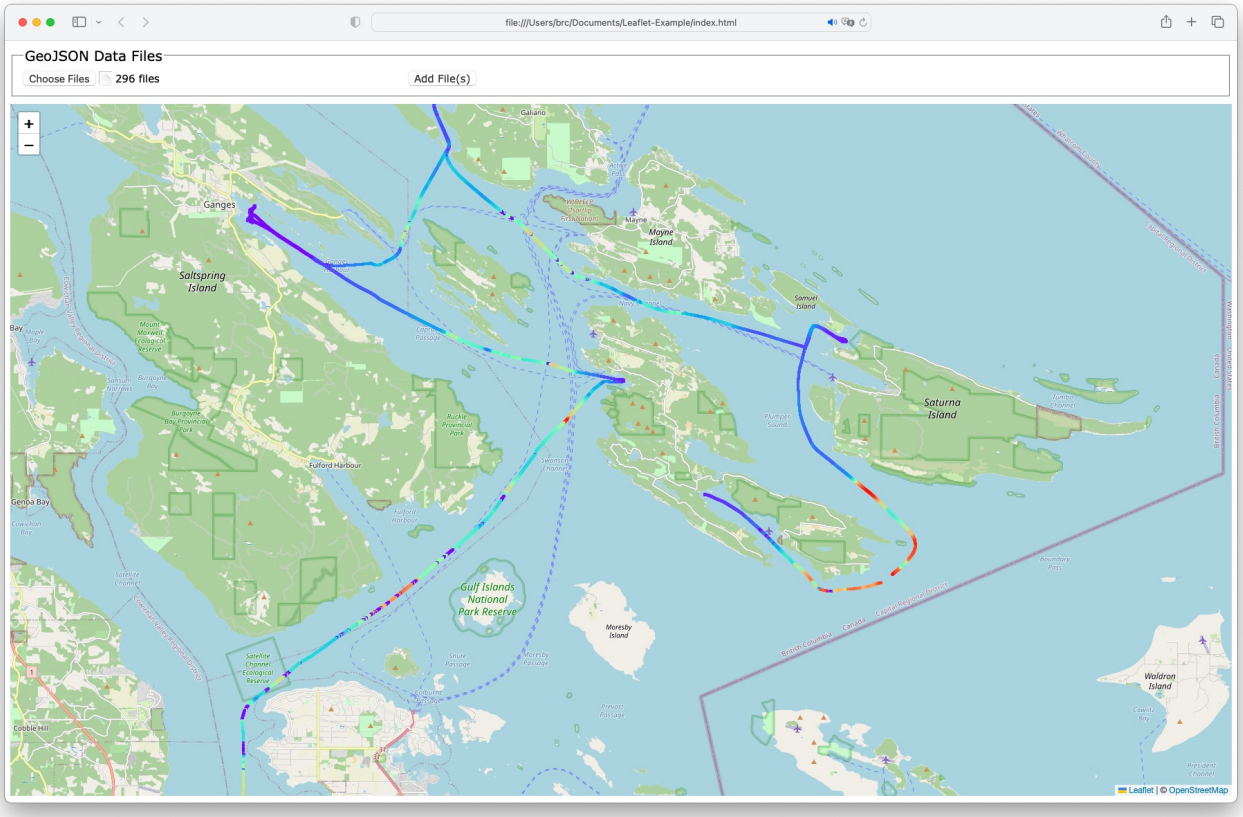
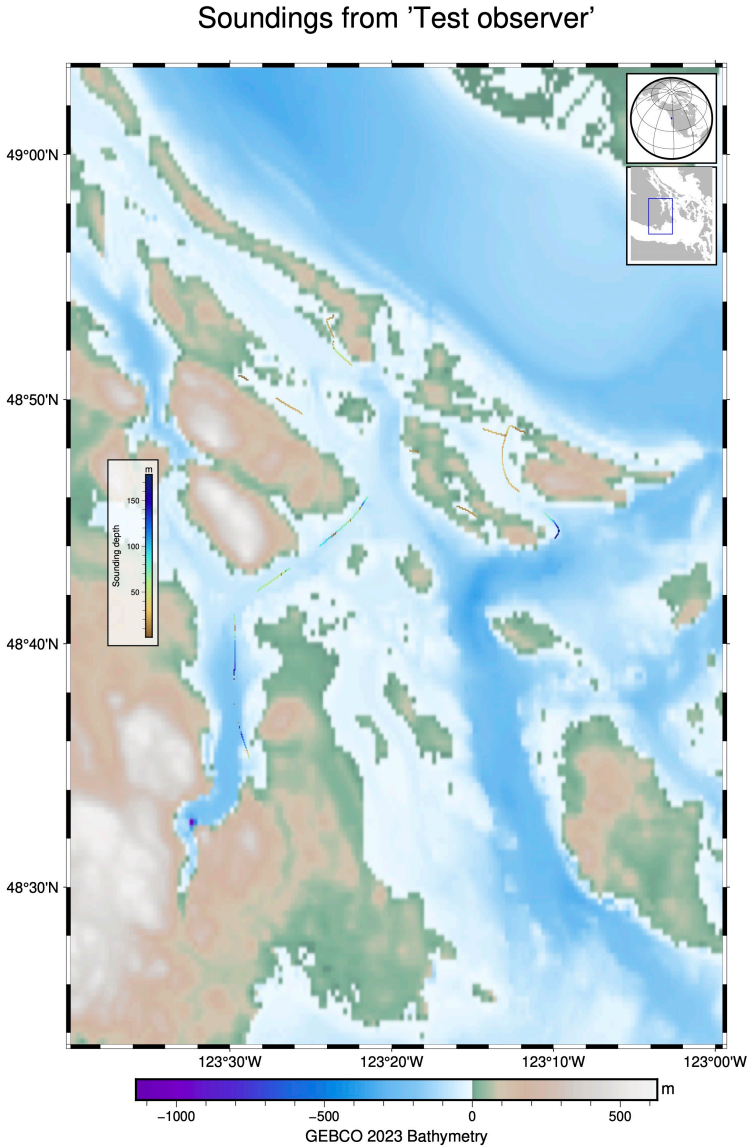
OpenVBI: An Open Source Toolbox for Volunteer Bathymetric Information



Brian R. Calder (brc@ccom.unh.edu)
Center for Coastal and Ocean Mapping & NOAA-UNH Joint Hydrographic Center
Chase Ocean Engineering Lab, 24 Colovos Road, Durham NH 03824


main.d28ynt8exokw88.amplifyapp.com
IHO
CSB Pointstore Dashboard
Data Centre for Digital Bathymetry
Report Date: 2026-02-24
Total Count: 2,008,966,225
Archive Dates:
2017-04-25 to 2026-02-21
16 orders in the last 30 days
Providers
Rosepoint 1,404,162,637 (70%)
AquaMap 167,140,243 (8%)
Carnival 154,810,211 (8%)
SeaKeepers 116,884,337 (6%)
PGS 54,342,457 (3%)
GLOS 32,270,188 (2%)
Growth Chart
Fiscal Year Table
Archive Growth
count (thousands)
2017 2018 2019 2020 2021 2022 2023 2024 2025 2026
2,008,966,225
328,003
H3 Index 842a30dfffffffff
Zoom to
Point Count
328003
Vermont
Conway
Auburn
Brunswick
Sebago Lake
Portland
Rutland
Laconia
Sanford
Claremont
Concord
Rochester
Dover
Manchester
Keene
Hampton
Amesbury
Newburyport
Haverhill
Nashua
Lowell
Ipswich
Gloucester
Greenfield
Gardner
Leominster
Massachusetts
Boston
Framingham
Worcester
Springfield
Brockton
Woonsocket
Plymouth
Taunton
Cape
Massachusetts Bay
Quabbin Reservoir
Rodgers Swell
Howell Swell
Franklin Swell
292
CHS, Esri, GEBCO, Garmin, NaturalVue | NOAA, NCEI, International Hydrographic Organization, Data Centre for Digital Bathymetry | Esri, TomTom, Garmin, SafeGraph, FAO, METI/NASA, USGS, EPA, NPS, USFWS
Powered by Esri


file:///Users/brc/Documents/Leaflet-Example/index.html
GeoJSON Data Files
Choose Files
296 files
Add File(s)
Ganges
Saltspring Island
Mayne Island
Samuel Island
Saturna Island
Fulford Harbour
Genoa Bay
Gulf Islands National Park Reserve
Moresby Island
Waldron Island
Leaflet | © OpenStreetMap

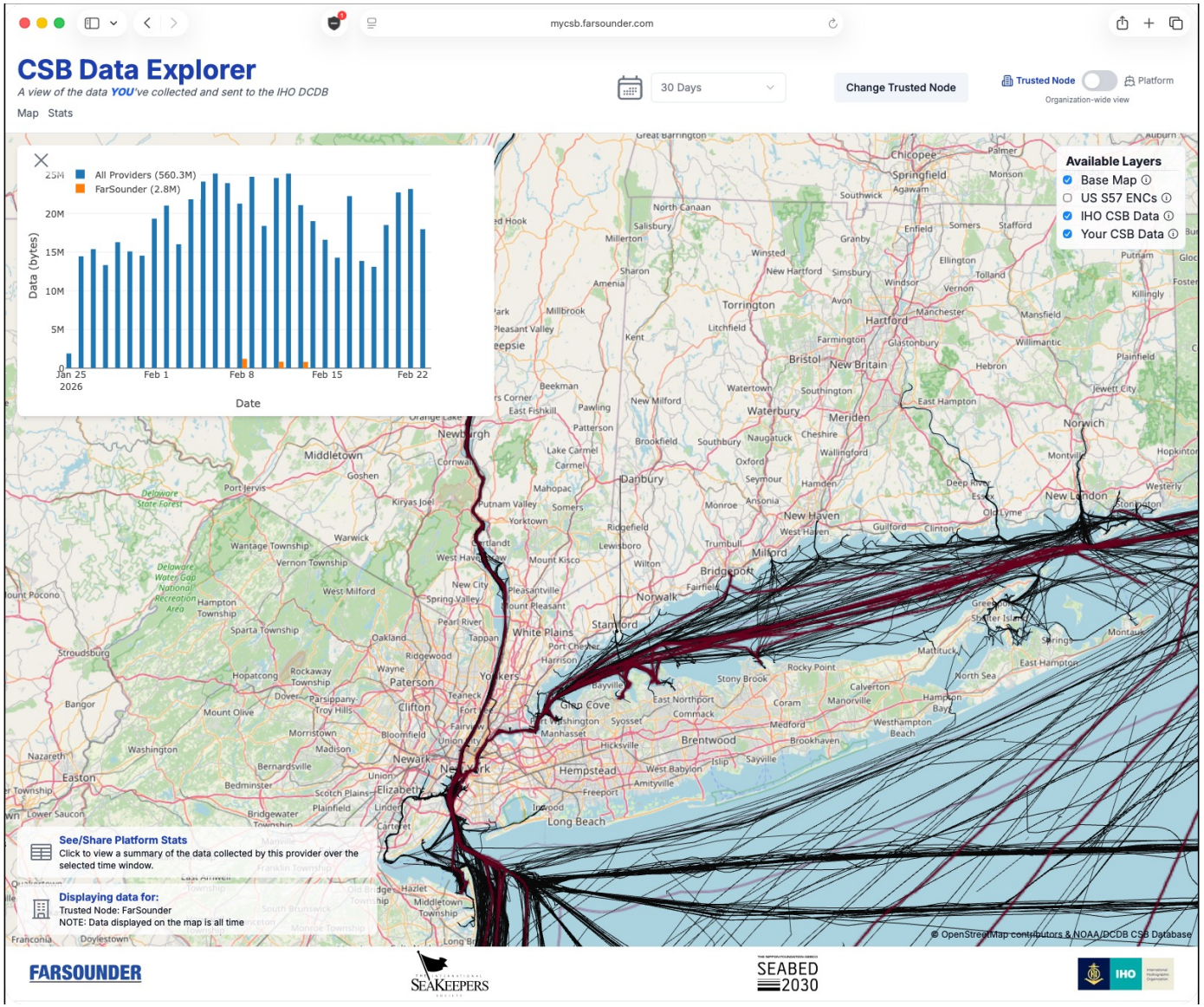

Soundings from 'Test observer'
49°00'N
48°50'N
48°40'N
48°30'N
123°30'W
123°20'W
123°10'W
123°00'W
Sounding depth
m
150
100
50
−1000
−500
0
500
m
GEBCO 2023 Bathymetry


GDAL


THE GENERIC MAPPING TOOLS


GEBCO


Wireless
Inexpensive
Bathymetry
Logger


mycsb.farsounder.com
CSB Data Explorer
A view of the data YOU've collected and sent to the IHO DCDB
Map Stats
30 Days
Change Trusted Node
Trusted Node
Platform
Organization-wide view
All Providers (560.3M)
FarSounder (2.8M)
Data (bytes)
25M
20M
15M
10M
5M
0
Jan 25
2026
Feb 1
Feb 8
Feb 15
Feb 22
Date
Available Layers
Base Map
US S57 ENCs
IHO CSB Data
Your CSB Data
See/Share Platform Stats
Click to view a summary of the data collected by this provider over the selected time window.
Displaying data for:
Trusted Node: FarSounder
NOTE: Data displayed on the map is all time
© OpenStreetMap contributors & NOAA/DCDB CSB Database
FARSOUNDER
SEAKEEPERS
SEABED 2030
IHO

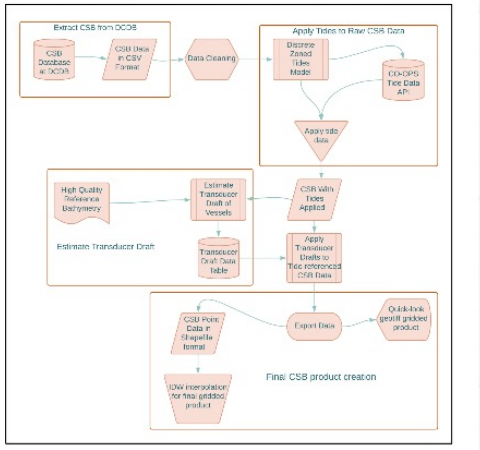


Figure 2: General workflow diagram for CSB processing script

Source: Klemm & Krabiel (US Hydro 2023)

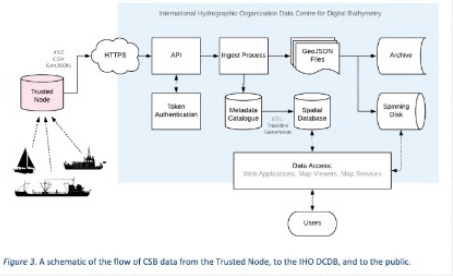
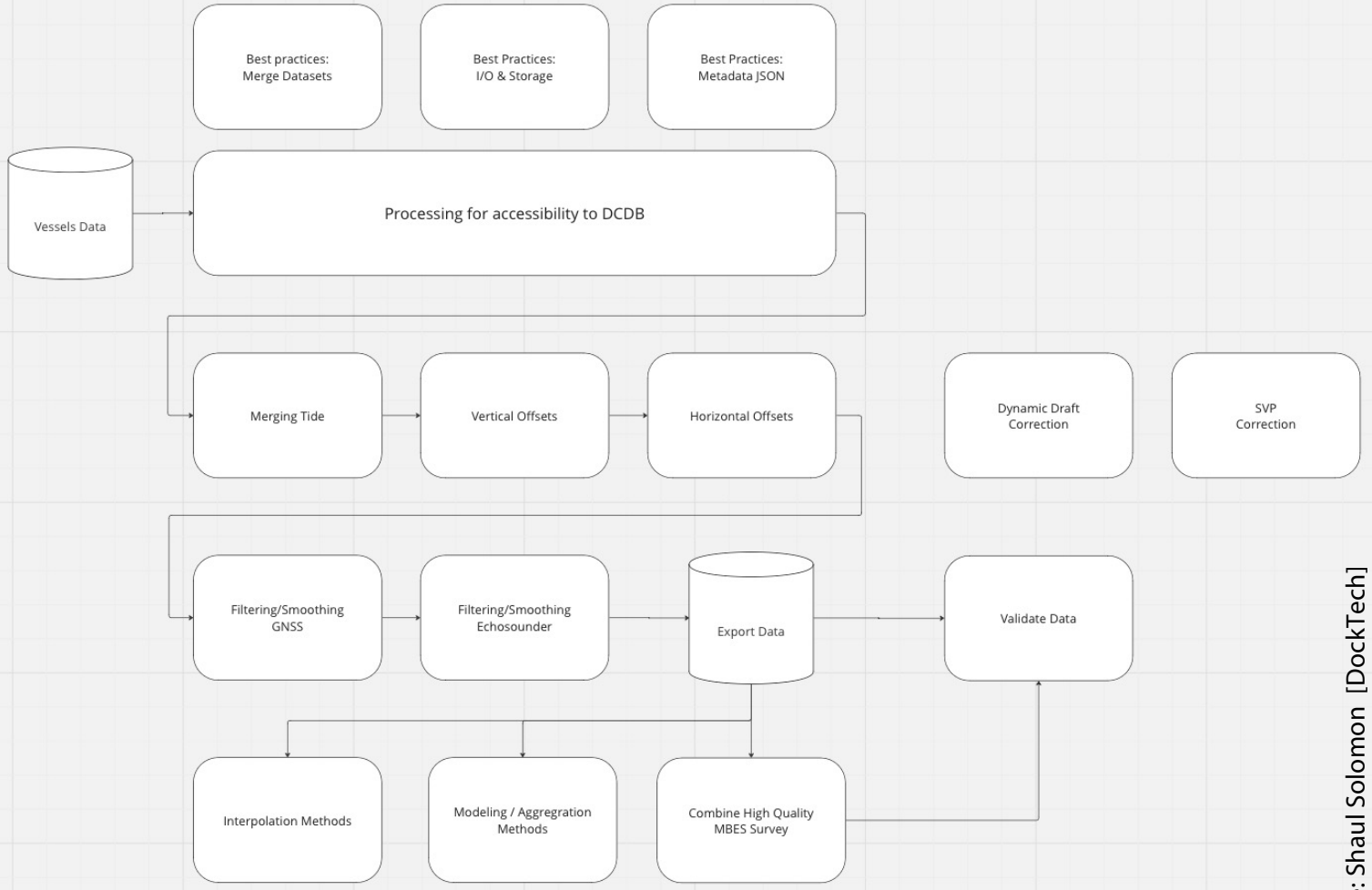


Figure 3. A schematic of the flow of CSB data from the Trusted Node, to the IHO DCDB, and to the public.

Source: DCDB



Source: Shaul Solomon [DockTech]

Why This Matters

Processing Consistency

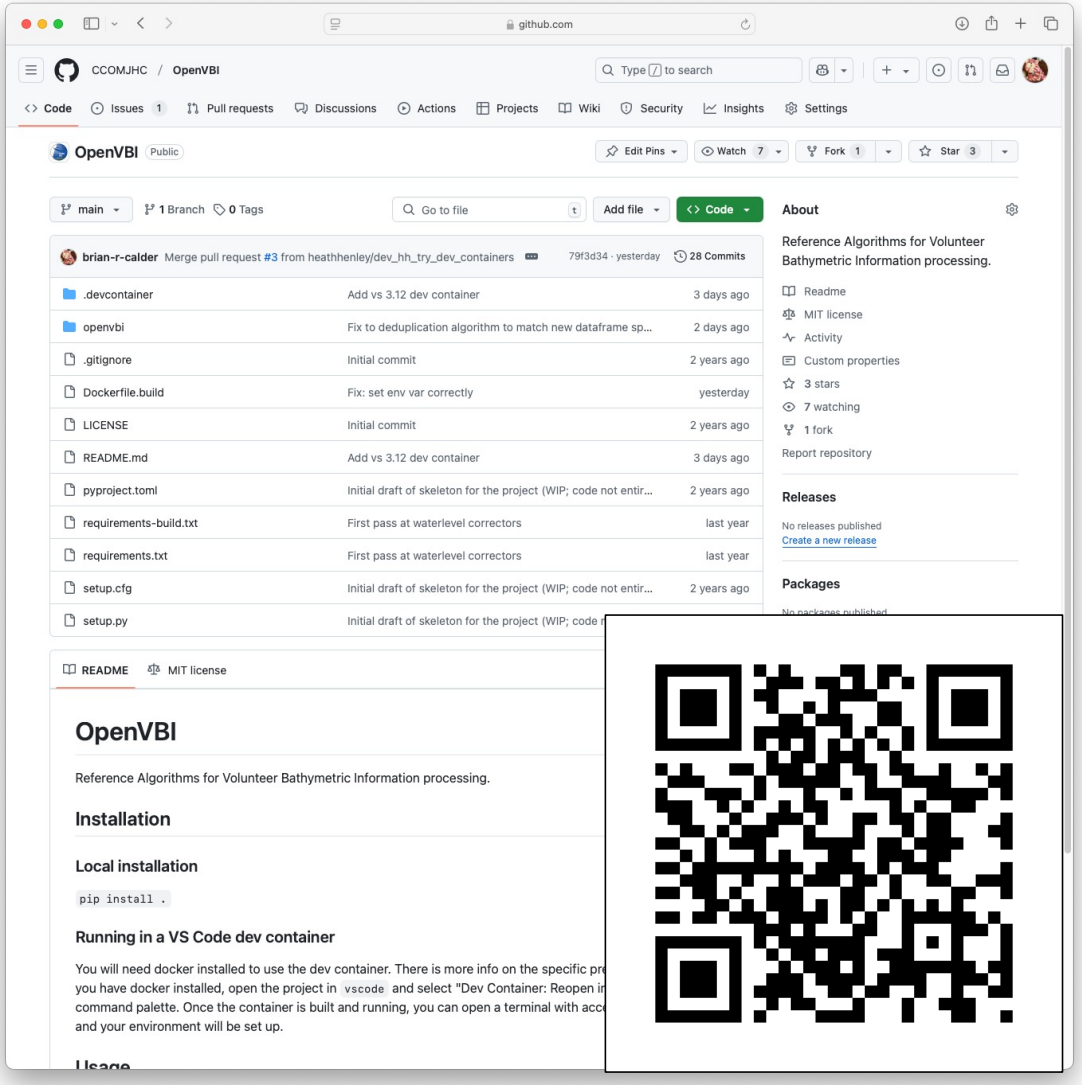
Metadata Construction

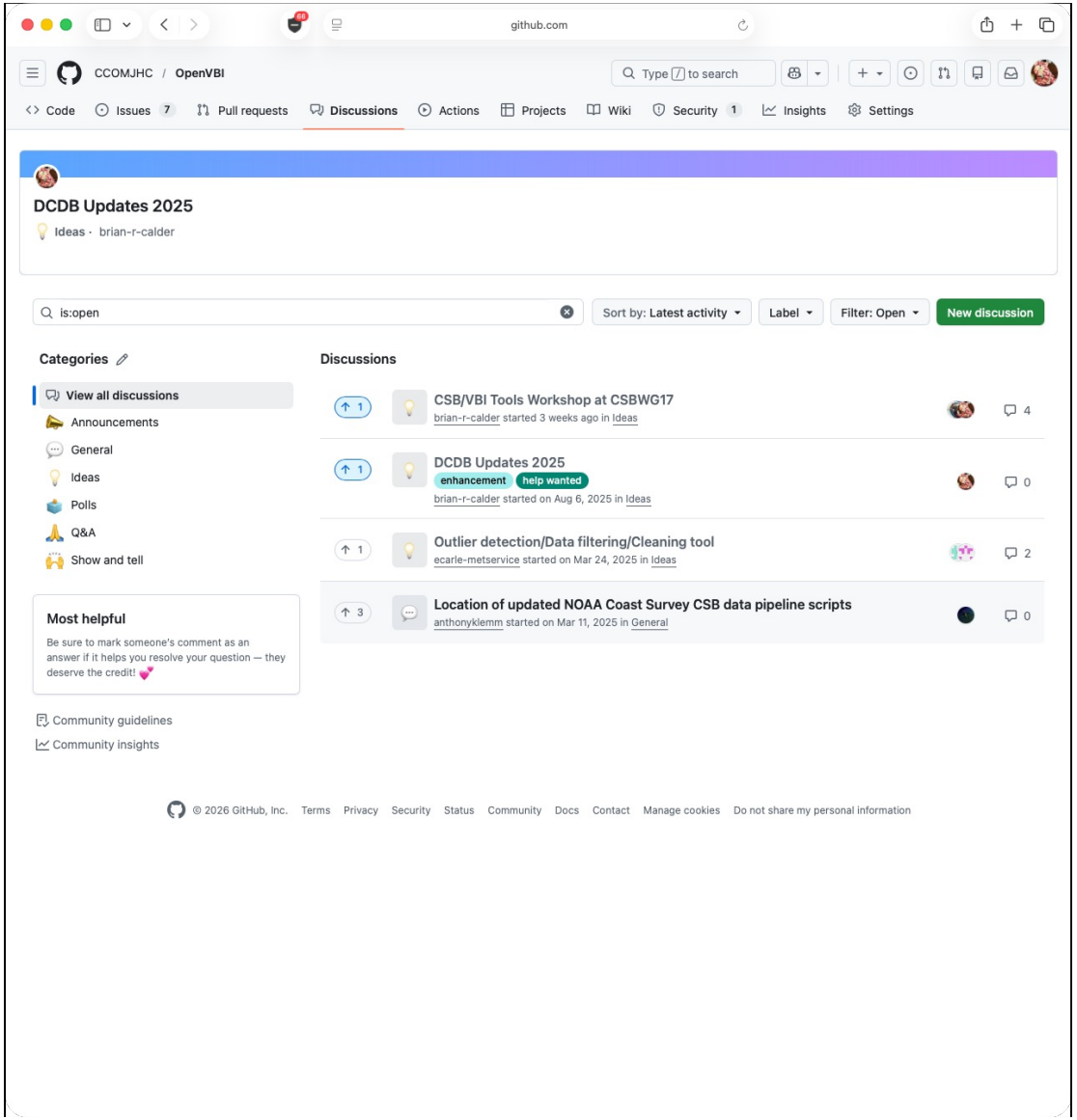
Archive Longevity

Engagement & Onboarding

OpenVBI

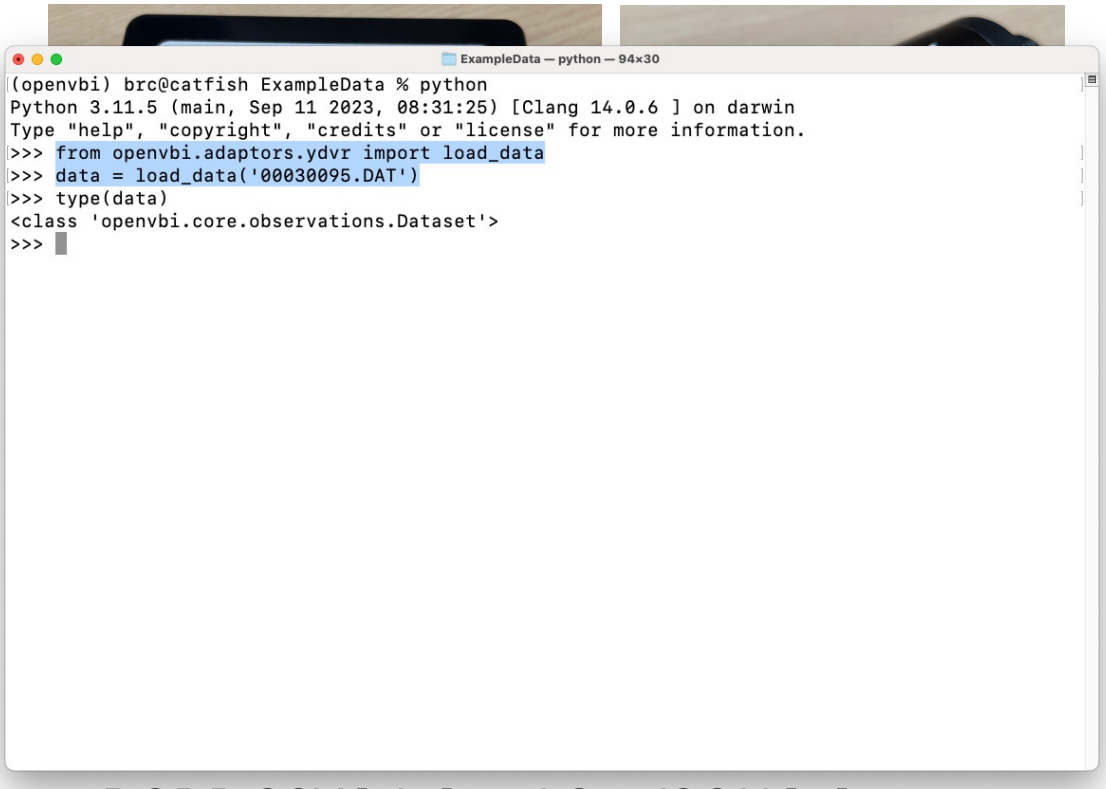
- Capture algorithms representing “**best practice**” from community.
- Document **workflow** for VBI/CSB data.
- Provide **strong metadata** management support.
- Cover whole VBI/CSB **data life cycle**: capture → processing → archive → product use.
- Encourage **collaboration** on common problems where possible.




github.com
CCOMJHC / OpenVBI
Type / to search
Code
Issues 7
Pull requests
Discussions
Actions
Projects
Wiki
Security 1
Insights
Settings
DCDB Updates 2025
Ideas · brian-r-calder
is:open
Sort by: Latest activity
Label
Filter: Open
New discussion
Categories
View all discussions
Announcements
General
Ideas
Polls
Q&A
Show and tell
Most helpful
Be sure to mark someone's comment as an answer if it helps you resolve your question — they deserve the credit!
Community guidelines
Community insights
Discussions
CSB/VBI Tools Workshop at CSBWG17
brian-r-calder started 3 weeks ago in Ideas
4
DCDB Updates 2025
enhancement
help wanted
brian-r-calder started on Aug 6, 2025 in Ideas
0
Outlier detection/Data filtering/Cleaning tool
ecarle-metservice started on Mar 24, 2025 in Ideas
2
Location of updated NOAA Coast Survey CSB data pipeline scripts
anthonyklemm started on Mar 11, 2025 in General
0
© 2026 GitHub, Inc.
Terms
Privacy
Security
Status
Community
Docs
Contact
Manage cookies
Do not share my personal information

OpenVBI

- Generic file adaptors (loggers, archive)
- Timestamp generation and interpolation of depth/position.
- Generic filtering for initial cleanup.
- Generic water level correction (and NOAA Single/Zone corrections)
- Rich metadata generation, management, and validation



+ DCDB CSV [r/w] and GeoJSON [w]

OpenVBI

- Generic file adaptors (loggers, archive)
- Timestamp generation and interpolation of depth/position.
- Generic filtering for initial cleanup.
- Generic water level correction (and NOAA Single/Zone corrections)
- Rich metadata generation, management, and validation

Ref. T (s)	Actual T
0.000	2025-03-11:105557.0
1.010	2025-03-11:105558.0

ExampleData — python — 94×30

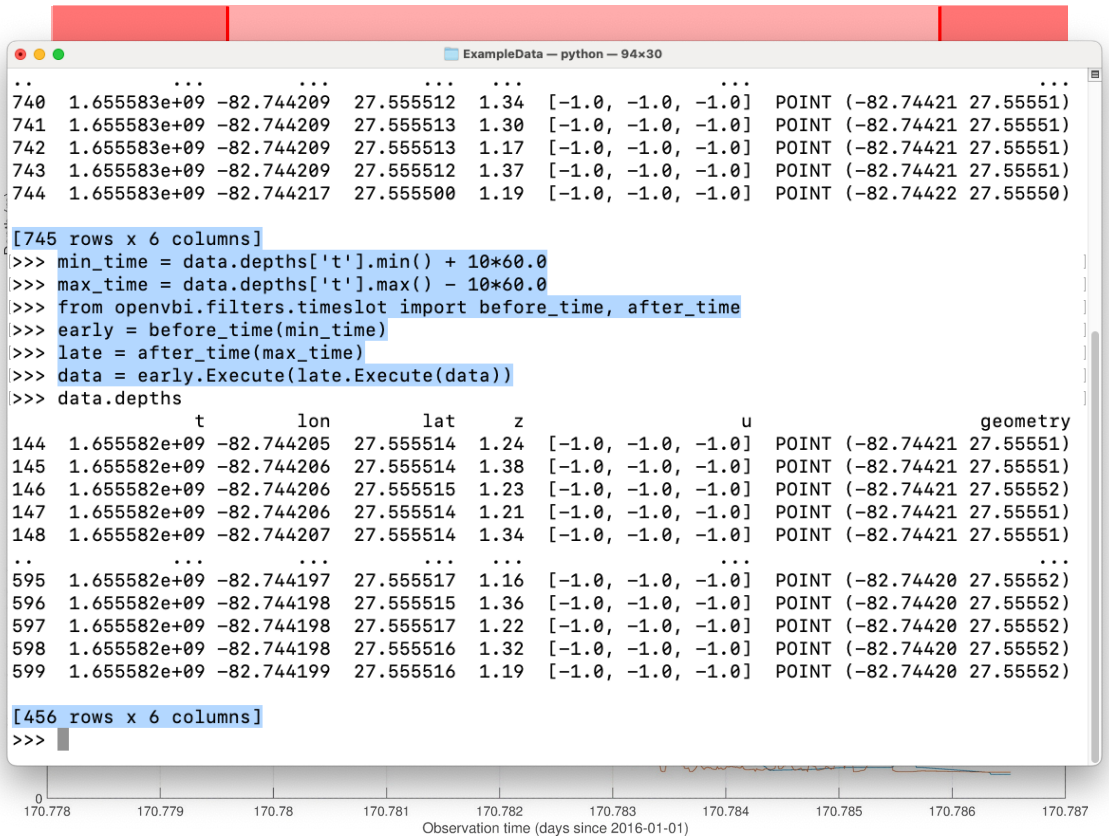
```
(openvbi) brc@catfish ExampleData % python
Python 3.11.5 (main, Sep 11 2023, 08:31:25) [Clang 14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from openvbi.adaptors.ydvr import load_data
>>> data = load_data('00030095.DAT')
>>> type(data)
<class 'openvbi.core.observations.Dataset'>
>>> data.generate_observations('Depth')
>>> type(data.depths)
<class 'geopandas.geodataframe.GeoDataFrame'>
>>> data.depths
                t        lon        lat     z                  u                     geometry
0    1.655582e+09 -82.744197  27.555516  1.32  [-1.0, -1.0, -1.0]  POINT (-82.74420 27.55552)
1    1.655582e+09 -82.744197  27.555517  1.29  [-1.0, -1.0, -1.0]  POINT (-82.74420 27.55552)
2    1.655582e+09 -82.744196  27.555517  1.17  [-1.0, -1.0, -1.0]  POINT (-82.74420 27.55552)
3    1.655582e+09 -82.744197  27.555516  1.35  [-1.0, -1.0, -1.0]  POINT (-82.74420 27.55552)
4    1.655582e+09 -82.744197  27.555517  1.25  [-1.0, -1.0, -1.0]  POINT (-82.74420 27.55552)
..            ...        ...        ...   ...                ...                        ...
740  1.655583e+09 -82.744209  27.555512  1.34  [-1.0, -1.0, -1.0]  POINT (-82.74421 27.55551)
741  1.655583e+09 -82.744209  27.555513  1.30  [-1.0, -1.0, -1.0]  POINT (-82.74421 27.55551)
742  1.655583e+09 -82.744209  27.555513  1.17  [-1.0, -1.0, -1.0]  POINT (-82.74421 27.55551)
743  1.655583e+09 -82.744209  27.555512  1.37  [-1.0, -1.0, -1.0]  POINT (-82.74421 27.55551)
744  1.655583e+09 -82.744217  27.555500  1.19  [-1.0, -1.0, -1.0]  POINT (-82.74422 27.55550)

[745 rows x 6 columns]
>>>
```

3.052	(-82.744197, 27.555518)
4.039	(-82.744203, 27.555535)

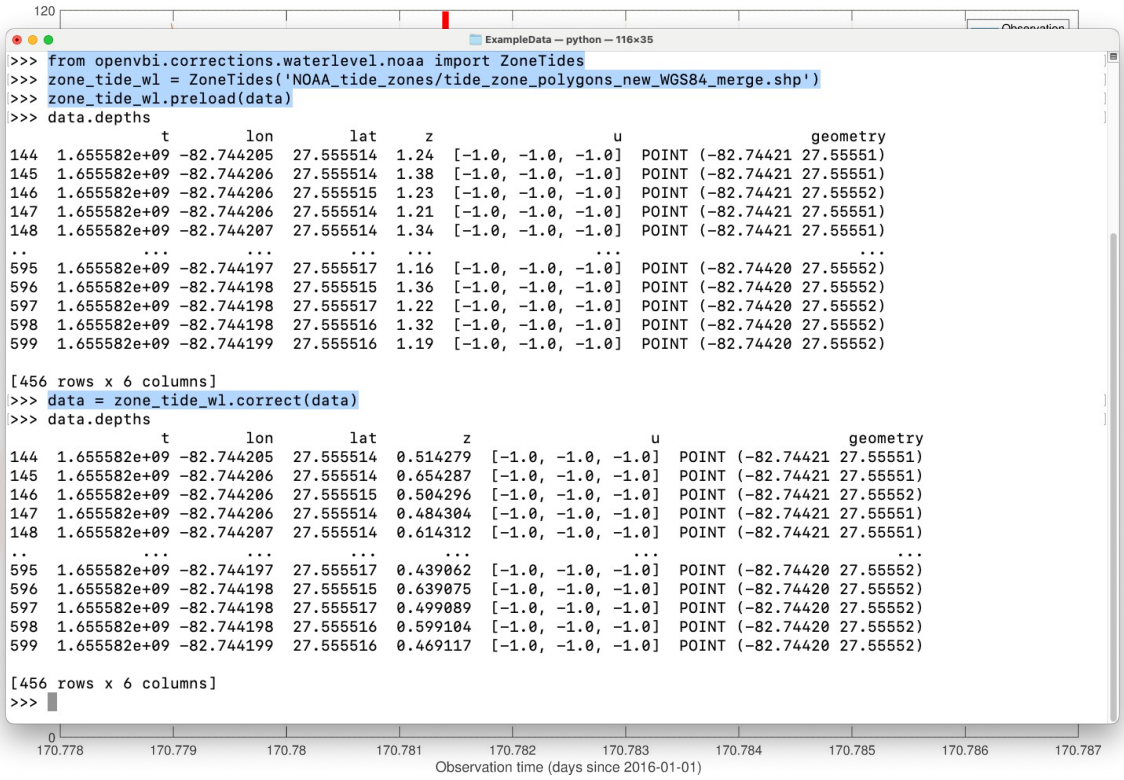
OpenVBI

- Generic file adaptors (loggers, archive)
- Timestamp generation and interpolation of depth/position.
- Generic filtering for initial cleanup.
- Generic water level correction (and NOAA Single/Zone corrections)
- Rich metadata generation, management, and validation



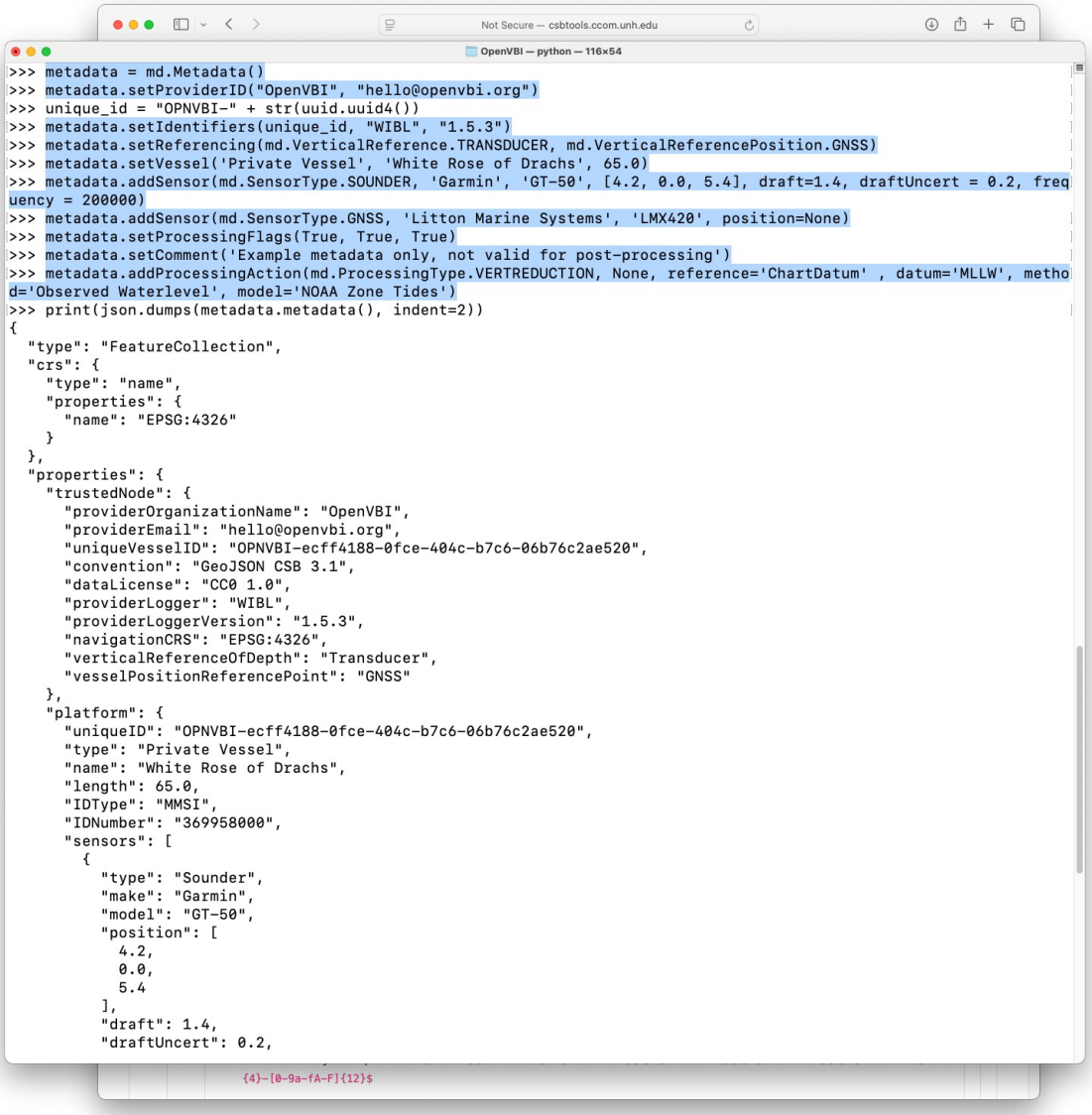
OpenVBI

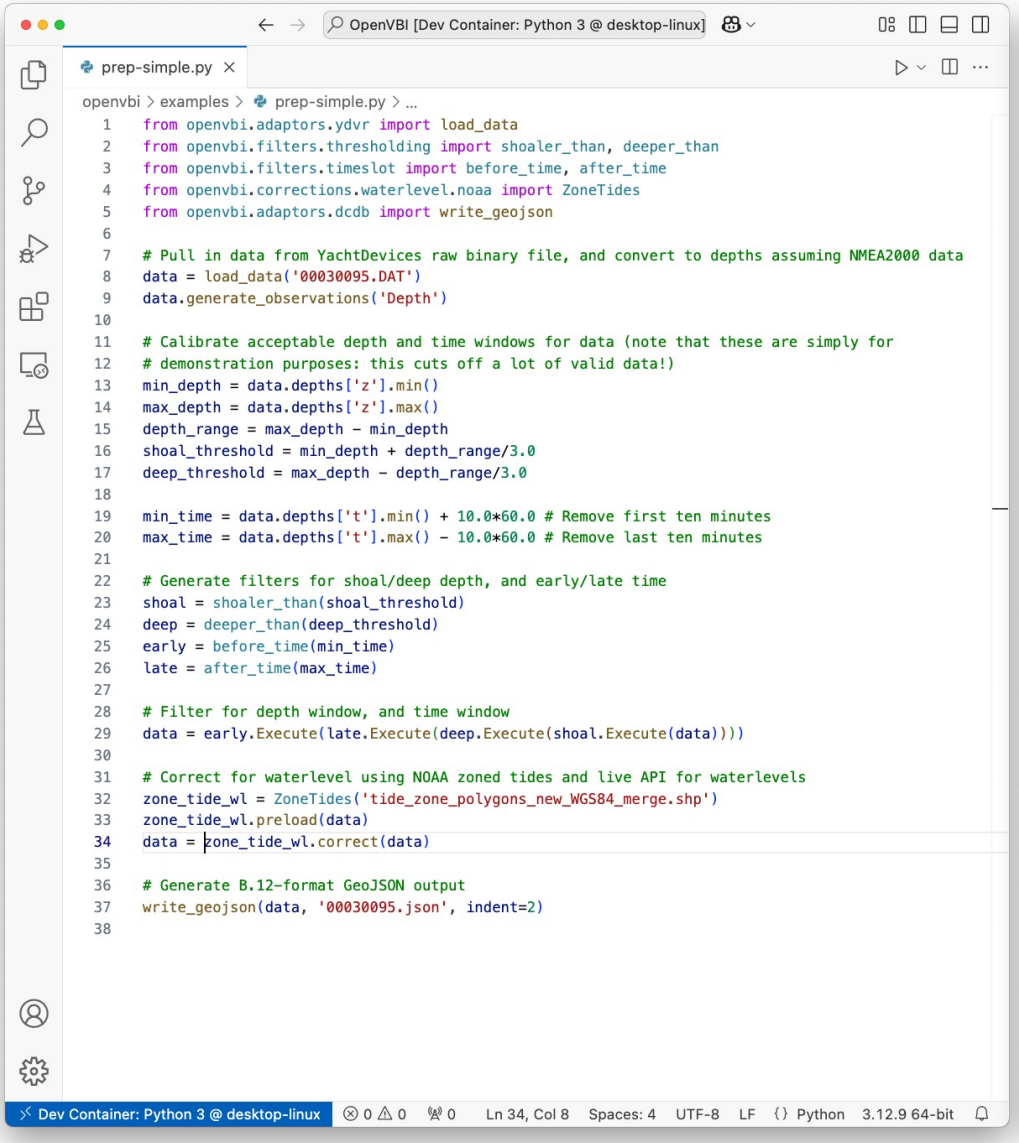
- Generic file adaptors (loggers, archive)
- Timestamp generation and interpolation of depth/position.
- Generic filtering for initial cleanup.
- Generic water level correction (and NOAA Single/Zone corrections)
- Rich metadata generation, management, and validation

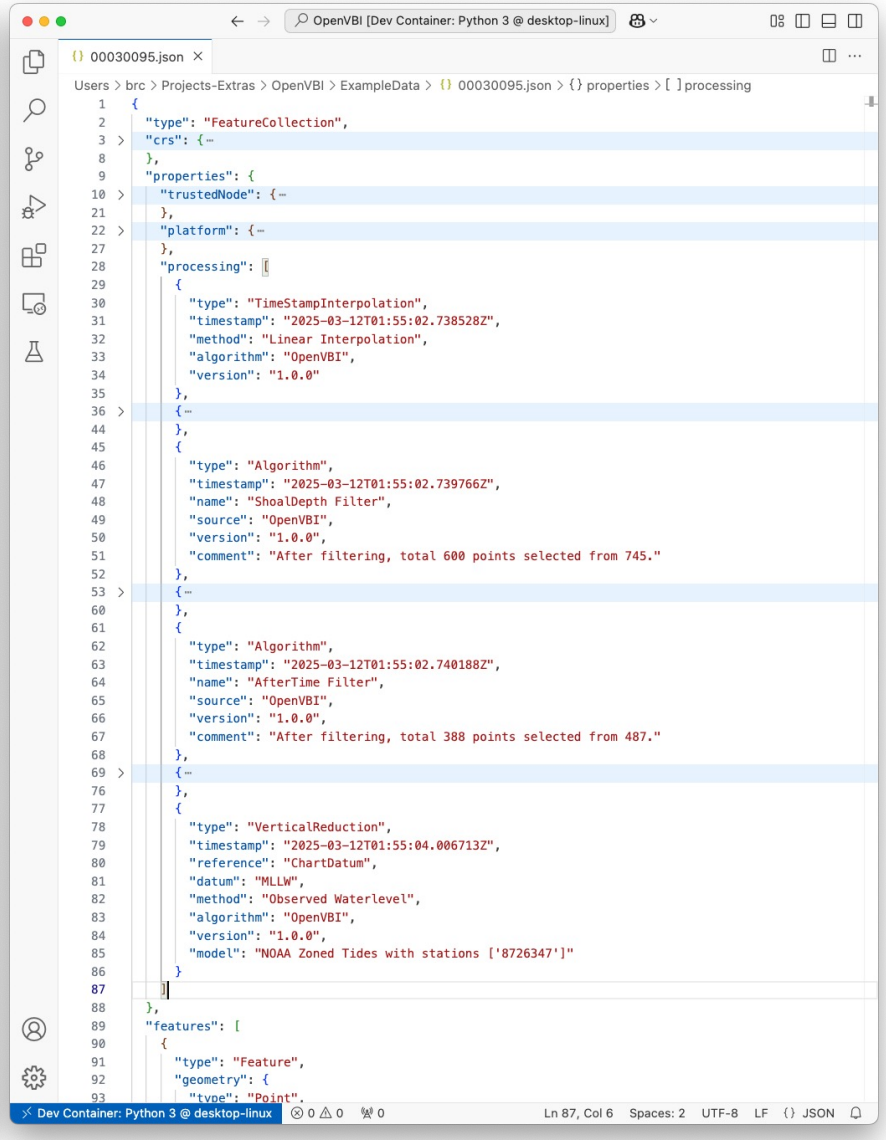


OpenVBI

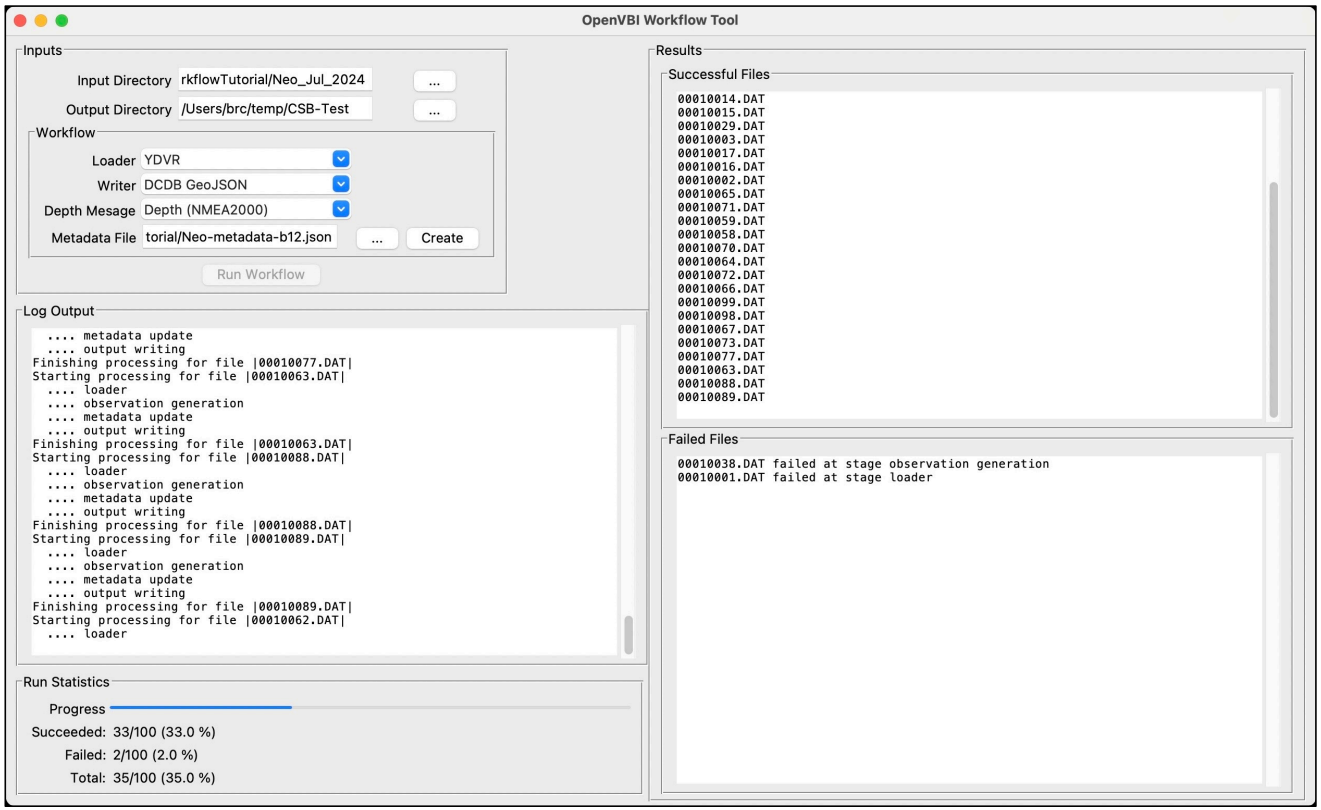
- Generic file adaptors (loggers, archive)
- Timestamp generation and interpolation of depth/position.
- Generic filtering for initial cleanup.
- Generic water level correction (and NOAA Single/Zone corrections)
- Rich metadata generation, management, and validation

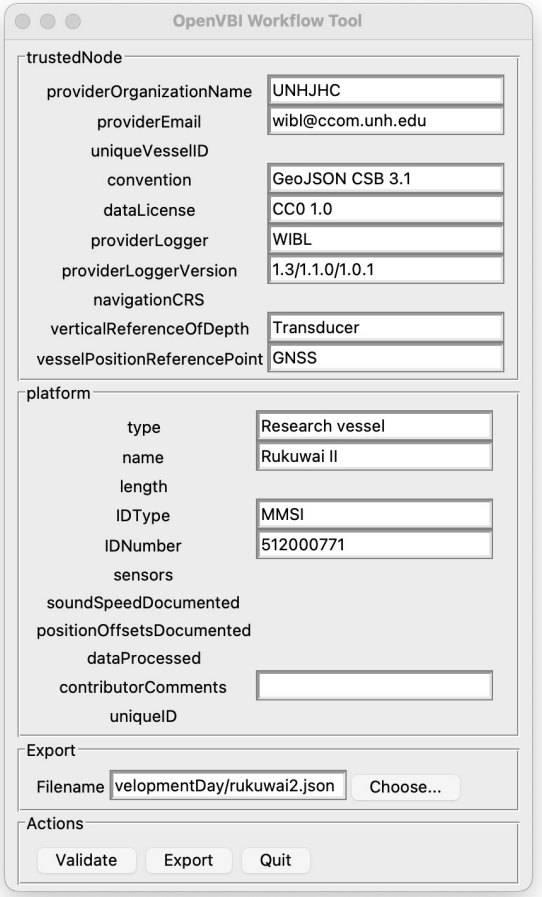



OpenVBI [Dev Container: Python 3 @ desktop-linux]
prep-simple.py
openvbi > examples > prep-simple.py > ...
from openvbi.adaptors.ydvr import load_data
from openvbi.filters.thresholding import shoaler_than, deeper_than
from openvbi.filters.timeslot import before_time, after_time
from openvbi.corrections.waterlevel.noaa import ZoneTides
from openvbi.adaptors.dcdb import write_geojson
# Pull in data from YachtDevices raw binary file, and convert to depths assuming NMEA2000 data
data = load_data('00030095.DAT')
data.generate_observations('Depth')
# Calibrate acceptable depth and time windows for data (note that these are simply for
# demonstration purposes: this cuts off a lot of valid data!)
min_depth = data.depths['z'].min()
max_depth = data.depths['z'].max()
depth_range = max_depth - min_depth
shoal_threshold = min_depth + depth_range/3.0
deep_threshold = max_depth - depth_range/3.0
min_time = data.depths['t'].min() + 10.0*60.0 # Remove first ten minutes
max_time = data.depths['t'].max() - 10.0*60.0 # Remove last ten minutes
# Generate filters for shoal/deep depth, and early/late time
shoal = shoaler_than(shoal_threshold)
deep = deeper_than(deep_threshold)
early = before_time(min_time)
late = after_time(max_time)
# Filter for depth window, and time window
data = early.Execute(late.Execute(deep.Execute(shoal.Execute(data))))
# Correct for waterlevel using NOAA zoned tides and live API for waterlevels
zone_tide_wl = ZoneTides('tide_zone_polygons_new_WGS84_merge.shp')
zone_tide_wl.preload(data)
data = zone_tide_wl.correct(data)
# Generate B.12-format GeoJSON output
write_geojson(data, '00030095.json', indent=2)
Dev Container: Python 3 @ desktop-linux
Ln 34, Col 8
Spaces: 4
UTF-8
LF
Python
3.12.9 64-bit


OpenVBI [Dev Container: Python 3 @ desktop-linux]
00030095.json
Users > brc > Projects-Extras > OpenVBI > ExampleData > 00030095.json > properties > processing
{
"type": "FeatureCollection",
"crs": {…
},
"properties": {
"trustedNode": {…
},
"platform": {…
},
"processing": [
{
"type": "TimeStampInterpolation",
"timestamp": "2025-03-12T01:55:02.738528Z",
"method": "Linear Interpolation",
"algorithm": "OpenVBI",
"version": "1.0.0"
},
{…
},
{
"type": "Algorithm",
"timestamp": "2025-03-12T01:55:02.739766Z",
"name": "ShoalDepth Filter",
"source": "OpenVBI",
"version": "1.0.0",
"comment": "After filtering, total 600 points selected from 745."
},
{…
},
{
"type": "Algorithm",
"timestamp": "2025-03-12T01:55:02.740188Z",
"name": "AfterTime Filter",
"source": "OpenVBI",
"version": "1.0.0",
"comment": "After filtering, total 388 points selected from 487."
},
{…
},
{
"type": "VerticalReduction",
"timestamp": "2025-03-12T01:55:04.006713Z",
"reference": "ChartDatum",
"datum": "MLLW",
"method": "Observed Waterlevel",
"algorithm": "OpenVBI",
"version": "1.0.0",
"model": "NOAA Zoned Tides with stations ['8726347']"
}
]
},
"features": [
{
"type": "Feature",
"geometry": {
"type": "Point",
Dev Container: Python 3 @ desktop-linux
Ln 87, Col 6
Spaces: 2
UTF-8
LF
JSON

OpenVBI


OpenVBI Workflow Tool
Inputs
Input Directory rkflowTutorial/Neo_Jul_2024 ...
Output Directory /Users/brc/temp/CSB-Test ...
Workflow
Loader YDVR
Writer DCDB GeoJSON
Depth Mesage Depth (NMEA2000)
Metadata File torial/Neo-metadata-b12.json ... Create
Run Workflow
Log Output
.... metadata update
.... output writing
Finishing processing for file |00010077.DAT|
Starting processing for file |00010063.DAT|
.... loader
.... observation generation
.... metadata update
.... output writing
Finishing processing for file |00010063.DAT|
Starting processing for file |00010088.DAT|
.... loader
.... observation generation
.... metadata update
.... output writing
Finishing processing for file |00010088.DAT|
Starting processing for file |00010089.DAT|
.... loader
.... observation generation
.... metadata update
.... output writing
Finishing processing for file |00010089.DAT|
Starting processing for file |00010062.DAT|
.... loader
Run Statistics
Progress
Succeeded: 33/100 (33.0 %)
Failed: 2/100 (2.0 %)
Total: 35/100 (35.0 %)
Results
Successful Files
00010014.DAT
00010015.DAT
00010029.DAT
00010003.DAT
00010017.DAT
00010016.DAT
00010002.DAT
00010065.DAT
00010071.DAT
00010059.DAT
00010058.DAT
00010070.DAT
00010064.DAT
00010072.DAT
00010066.DAT
00010099.DAT
00010098.DAT
00010067.DAT
00010073.DAT
00010077.DAT
00010063.DAT
00010088.DAT
00010089.DAT
Failed Files
00010038.DAT failed at stage observation generation
00010001.DAT failed at stage loader


OpenVBI Workflow Tool
trustedNode
providerOrganizationName UNHJHC
providerEmail wibl@ccom.unh.edu
uniqueVesselID
convention GeoJSON CSB 3.1
dataLicense CC0 1.0
providerLogger WIBL
providerLoggerVersion 1.3/1.1.0/1.0.1
navigationCRS
verticalReferenceOfDepth Transducer
vesselPositionReferencePoint GNSS
platform
type Research vessel
name Rukuwai II
length
IDType MMSI
IDNumber 512000771
sensors
soundSpeedDocumented
positionOffsetsDocumented
dataProcessed
contributorComments
uniqueID
Export
Filename velopmentDay/rukuwai2.json Choose...
Actions
Validate Export Quit

What's Next?

OUTLIER DETECTION & REMOVAL

PRODUCT GENERATION

VERTICAL OFFSET ESTIMATION & CORRECTION

GLOBAL WATERLEVEL CORRECTION

UNCERTAINTY ESTIMATION

FULL PROCESSING WORKFLOWS

Brian R. Calder (brc@ccom.unh.edu, +1-603-862-0526)
Research Professor & Associate Director

Center for Coastal and Ocean Mapping & NOAA-UNH Joint Hydrographic Center
Chase Ocean Engineering Lab, University of New Hampshire
24 Colovos Road
Durham, NH 03824
USA

Sponsored by NOAA Grants NA20NOS4000196 and NA25NOSX400C0001,
“Continuation of the Joint Hydrographic Center”