

Data Flagging Dashboard

Main Data Chain Jupyter Notebook

The screenshot shows a JupyterLab interface with a file browser on the left and a notebook editor on the right. The notebook is titled "demo_acsm_pipeline.ipynb" and is running on "localhost:8889". The notebook content includes a section titled "Step 3: Perform QC/QA Analysis" with the following steps:

- Generate automated flags based on validity thresholds for diagnostic channels.
- (Optional) Generate manual flags using the **Data Flagging App**, accessible at: <http://localhost:8050/>
- (Optional) Inspect Previously Generated Flags for Correctness using the Jupyter Notebook workflow available at: http://localhost:8889/lab/tree/notebooks/demo_visualize_diagnostic_flags_from_hdf5_file.ipynb

The code cell [7] contains the following Python code:

```
[7]: dataset_name = f'ACSM_TOFWARE/{YEAR}/ACSM_{STATION_ABBR}_{YEAR}_meta.t  
path_to_config_file = 'pipelines/params/validity_thresholds.yaml'  
#command = ['python', 'pipelines/steps/compute_automated_flags.py', pa  
#status = subprocess.run(command, capture_output=True, check=True)  
#print(status.stdout.decode())  
path_to_data_file = CAMPAIGN_DATA_FILE  
generate_flags(path_to_data_file, 'diagnostics', capture_renku_metadata
```

The output of the code cell shows:

```
Total rows: 104669  
NaT (missing) values: 0  
Percentage of data loss: 0.0000%
```

The screenshot shows the Data Flagging Dashboard interface, which is a web application running on "localhost:8050". The dashboard includes the following sections:

- Open HDF5 file**: A section for uploading or selecting an HDF5 file.
- Select data table**: A section for selecting a data table from a list of available tables.
- Create flags**: A section for creating flags, with a dropdown menu for "Invalidated by data originator" and buttons for "Create Flag", "Reset Flag", and "Commit Flag".
- ACSM_TOFWARE/2024 : Target and Diagnostic Channels**: A section displaying four time series plots for different channels: CH_11000, NH4_11000, NO3_11000, and Org_11000. Each plot shows data from January 2024 to November 2024.
- Enable Flag Visualization**: A section for enabling flag visualization, with a "Load Flags" button and a table of flags.

The table of flags has the following columns: id, startdate, and end. It contains 20 rows of data, with the first row being:

id	startdate	end
10	2024-06-26 07:24:24.4552	2024-06-26 07:24:24.4552
11	2024-11-29 13:28:02.5791	2024-11-29 13:28:02.5791
12	2024-03-03 06:56:02.0653	2024-03-03 06:56:02.0653
14	2024-02-21 18:46:20.4114	2024-02-21 18:46:20.4114
15	2024-02-23 15:20:28.6085	2024-02-23 15:20:28.6085
16	2024-03-03 10:13:34.4017	2024-03-03 10:13:34.4017
17	2024-03-07 12:08:44.304	2024-03-07 12:08:44.304
18	2024-03-09 11:53:23.8804	2024-03-09 11:53:23.8804
19	2024-03-11 15:40:57.1144	2024-03-11 15:40:57.1144
20	2024-05-27 16:47:28.625	2024-05-27 16:47:28.625

Visualization Jupyter Notebook

The screenshot shows the Visualization Jupyter Notebook interface, which is a web application running on "localhost:8889". The notebook is titled "demo_visualize_diagnostic_flags_from_hdf5_file.ipynb" and is running on "localhost:8889". The notebook content includes a section titled "VaporizerTemp_C over t_base" with the following plots:

- VaporizerTemp_C over t_base**: A line plot showing VaporizerTemp_C (Y-axis, 400 to 600) over time (X-axis, Jan 2024 to Nov 2024). The plot includes a blue line for VaporizerTemp_C and red vertical bars for Invalid Region.
- FlowRate_ccs over t_base**: A line plot showing FlowRate_ccs (Y-axis, 0 to 3) over time (X-axis, Jan 2024 to Nov 2024). The plot includes a blue line for FlowRate_ccs and red vertical bars for Invalid Region.
- FilamentEmission_mA over t_base**: A line plot showing FilamentEmission_mA (Y-axis, 0.4 to 1) over time (X-axis, Jan 2024 to Nov 2024). The plot includes a blue line for FilamentEmission_mA and red vertical bars for Invalid Region.