

# CampaignDescriptor.YAML

```
input_file_directory: '/mnt/network/lac_ord/Data/JFJ/'
output_file_directory: '../data/'

project: 'Building FAIR data chains for atmospheric observations in the ACTRIS Switzerland Network'
contact: 'LeilaS'
group_id: '5502' # APOG

experiment: 'acsm_campaign' # beamtime, smog_chamber, lab_experiment
dataset_startdate:
dataset_enddate:
actris_level: '1'

# Define an (output) filename format of a resulting hdf5 file
station_abbr : 'JFJ' # Set this value as either "JFJ" or "PAY"
year : '2024'
instrument_name : 'ACSM-017' #'ACSM-092'
filename_format : "station_abbr, year"

instrument_datafolder:
- "ACSM_TOFWARE/2024"

integration_mode: 'collection'
# Specify datetimes (YYYY-MM-DD HH-MM-SS) at which experimental steps were created.
datetime_steps: []

# ACSM data chain
originator:
name: Simon, Leila
email: leila.simon@psi.ch
affiliation: Paul Scherrer Institute, PSI
department: Laboratory of Atmospheric Chemistry
address_line_1: ""
address_line_2: ""
postal_code: 5232
city: Villigen PSI
country: Switzerland
```

## Main Data Chain Jupyter Notebook

1.

The screenshot shows the JupyterLab interface for the 'demo\_acsm\_pipeline.ipynb' notebook. The left sidebar displays a file explorer with a tree view of the project structure, including folders like 'app', 'data', 'dima', 'figures', 'notebooks', 'pipelines', 'third\_party', and 'workflows'. The 'notebooks' folder is selected, showing a list of files including 'campaignDescriptor.yaml'. The main area displays the notebook content, which includes a title 'ACSM Data Chain Workflow', a list of steps, and a code cell for importing libraries and setting up the project root directory.

### ACSM Data Chain Workflow

In this notebook, we will go through our **ACSM Data Chain**. This involves the following steps:

1. Run the data integration pipeline to retrieve ACSM input data and prepare it for processing.
2. Perform QC/QA analysis.
3. (Optional) Conduct visual analysis for flag validation.
4. Prepare input data and QC/QA analysis results for submission to the EBAS database.

### Import Libraries and Data Chain Steps

- Execute (or Run) the cell below.

```
[1]: import sys
import os
# Set up project root directory

notebook_dir = os.getcwd() # Current working directory (assumes running from notebooks/)
project_path = os.path.normpath(os.path.join(notebook_dir, "..")) # Move up to project root
dima_path = os.path.normpath(os.path.join(project_path, "dima")) # Move up to project root

if project_path not in sys.path: # Avoid duplicate entries
    sys.path.append(project_path)
if dima_path not in sys.path:
    sys.path.insert(0, dima_path)
import subprocess

for item in sys.path:
    print(item)
```

2.

The screenshot shows the HDFView 3.3.1 application displaying the contents of the 'collection\_JFJ\_2024\_20250610.H5' file. The left pane shows a tree view of the file's structure, including folders like 'ACSM\_TOFWARE' and '2024'. The right pane displays the 'Object Attribute Info' for the selected object, showing a table of attributes with their names and types.

| Name         | Type   |
|--------------|--|
| actris_level | String, length = variable, padding = H5T_STF |
| contact      | String, length = variable, padding = H5T_STF |
| experiment   | String, length = variable, padding = H5T_STF |
| project      | String, length = variable, padding = H5T_STF |