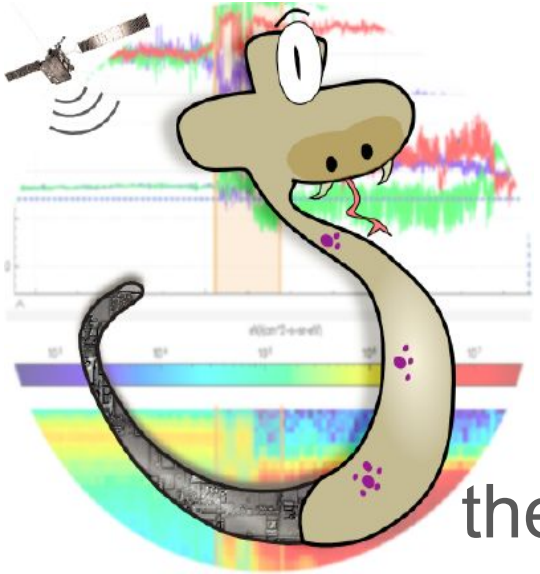


ATST
Action Thématique Soleil-Terre

SciQLop:

the **S**pace **P**hysicist's **IDE** **R**evolution

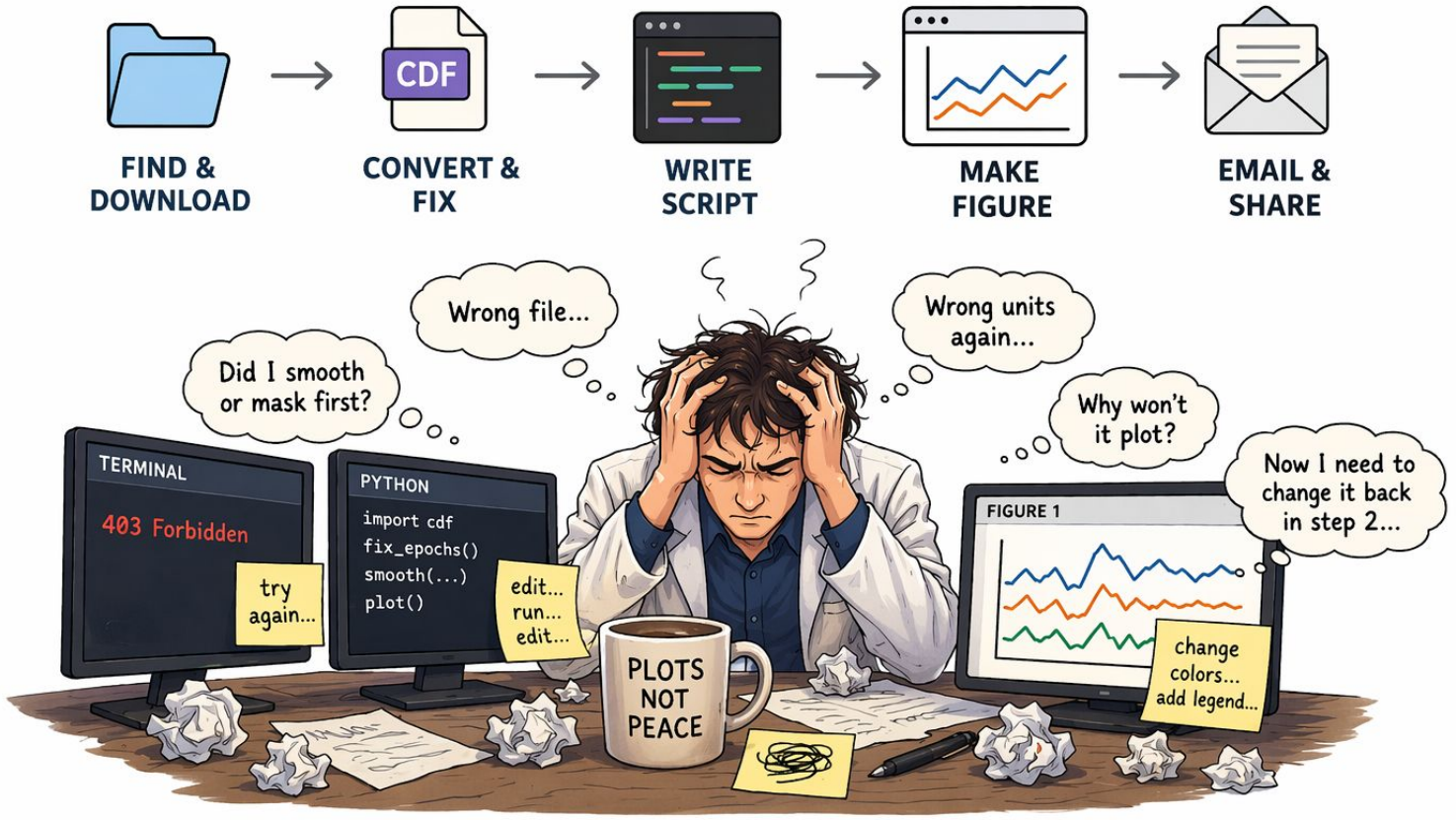
from archive to science, without leaving your environment

Colloque ATST 2026

Alexis Jeandet[1], Nicolas Aunai[1], Bruno Katra[1], Lina Hadid[1], Benjamin Renard[2], Myriam Bouchemit[2], Nicolas André[2], Vincent Génot[2], Christian Jacquey[2], Alexis Rouillard[2]

LPP irap astrophysique & planétologie cnes CDPP

Turning data into a plot shouldn't feel like this.



Science is hard enough. Your tools shouldn't be.

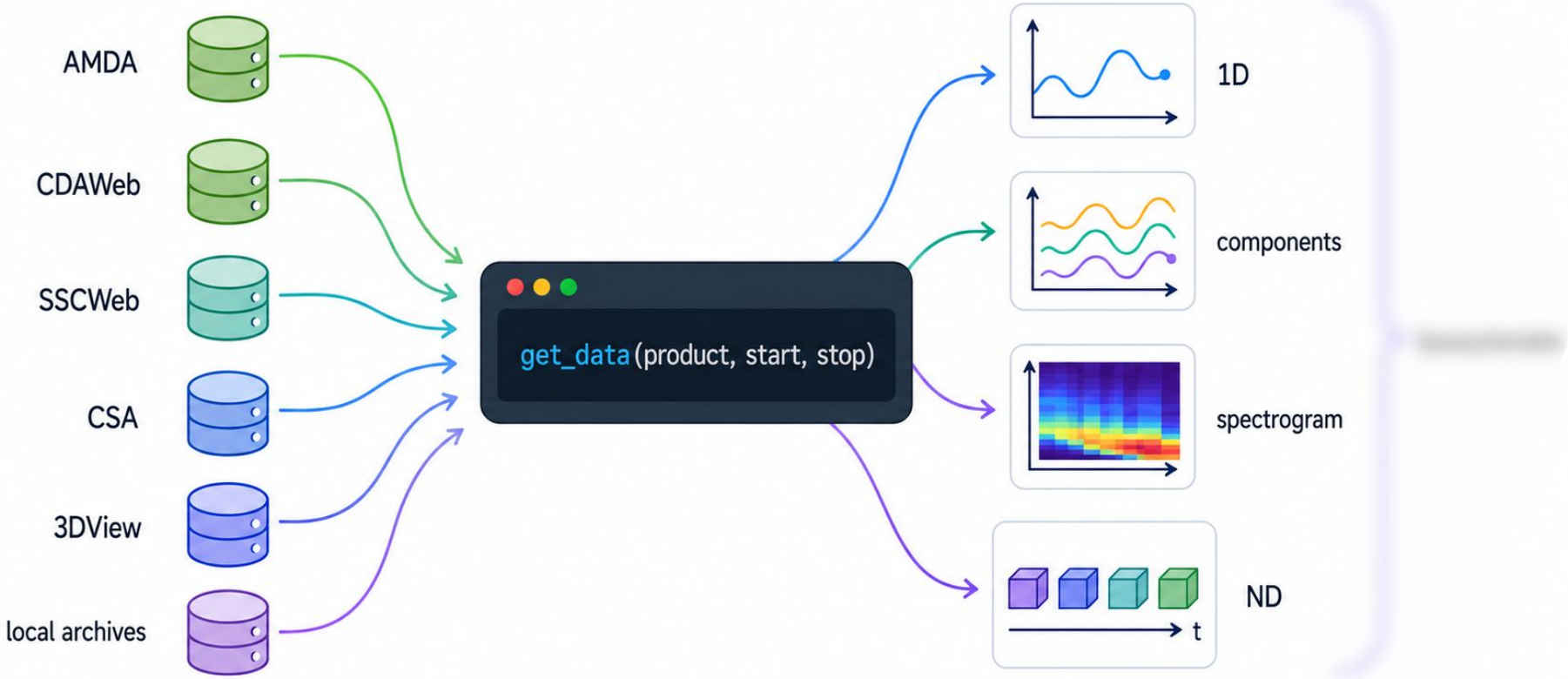
A lot of data in many places

- From **~60 years** of continuous measurements
- ~ **70 space** missions
- **~500TB** of measurements
- Across ~10 different servers
- Many differents instruments/products/modes
- Few files formats (CDF, NetCDF, Fits,...)
- Few standards

A lot of use cases

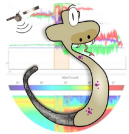
- Quick look?
- Look at an arbitrary set of data?
- Find/discover data?
- Multi-mission analysis?
- Massive statistic study?
- Instrument calibration?
- ...

The dream API



The ecosystem

SciQLop
the IDE, browse, visualise, script, annotate, share



Plugins

AppStore

AstraLint
data quality, validation, standards compliance

AstraLint

pip install astralint

Speasy
unified access, AMDA, CDA, CSA, SSC…

Speasy

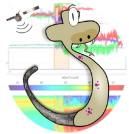
pip install speasy

PyCDF++
super-fast C++ CDF reader

pip install pycdfpp

The ecosystem

SciQLop
the IDE, browse, visualise, script, annotate, share



Plugins

AppStore

AstraLint
data quality, validation, standards compliance

pip install astralint

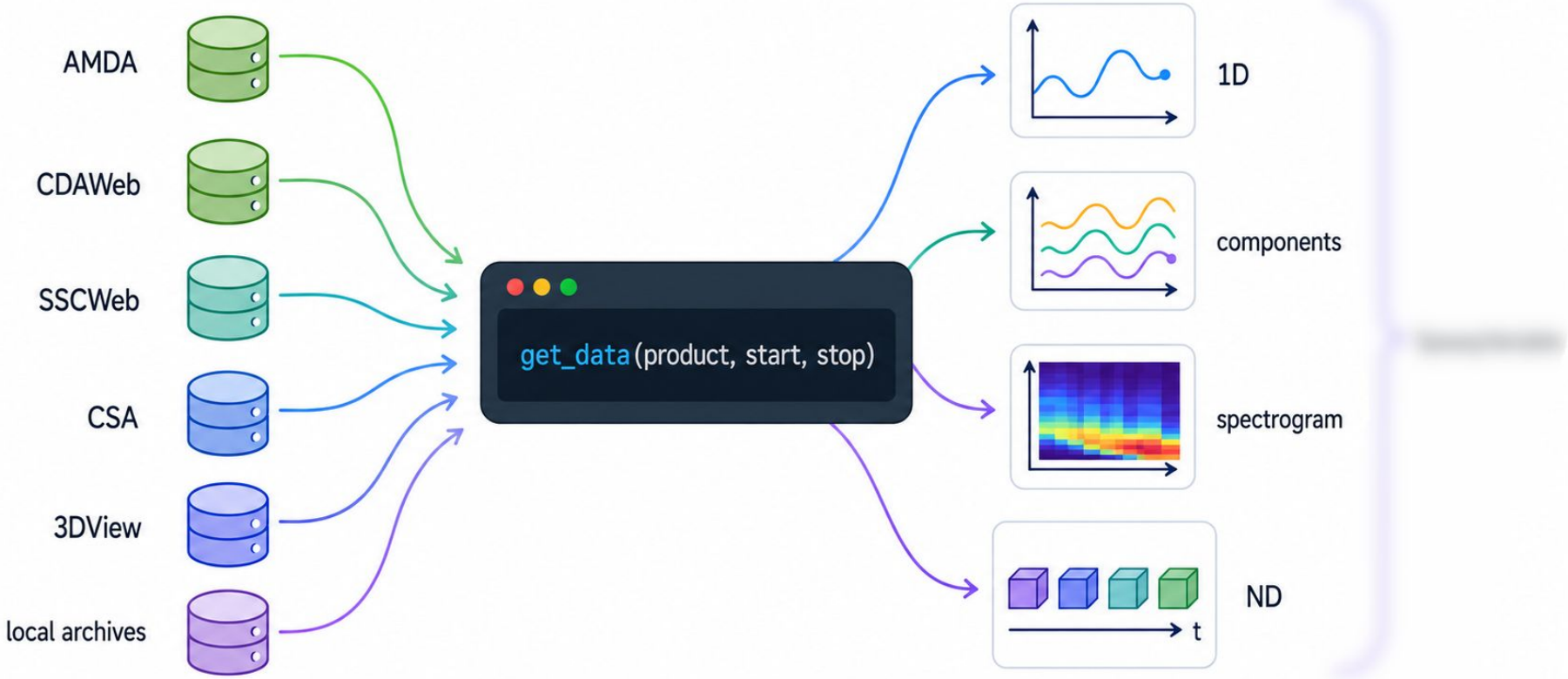
Speasy
unified access, AMDA, CDA, CSA, SSC…

pip install speasy

PyCDF++
super-fast C++ CDF reader

pip install pycdfpp

The dream API



Speasy: one API to rule them all

Speasy: one API to rule them all

70+ missions
AMDA · CDA · CSA · SSC · local

65,000+ products
one call, any archive

Tab-completable
discover data in IPython

Get

```
import speasy as spz
mag = spz.get_data(
    'amda/imf',
    "2016-6-2",
    "2016-6-5")
```

-> **SpeasyVariable**

No boilerplate

Compute

```
import speasy as spz
import numpy as np

mag_total =
np.linalg.norm(mag,
axis=1)
```

-> **SpeasyVariable**

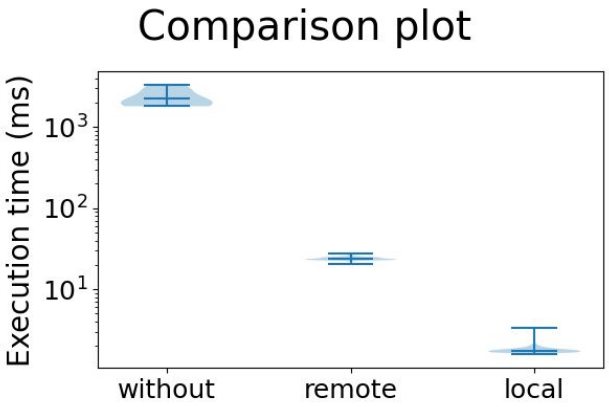
NumPy just works

Align

```
from speasy.signal import
resampling

Tperp_interp, Tpara_interp =
resampling.interpolate(
    b, [Tperp, Tpara])
```

-> on **b**'s time axis

scipy-compatible, time-aware



Super fast access to data

your code

local cache

shared public cache (LPP)

AMDA

CDAWeb

CSA

[20µs:10ms] [1ms:1s] [100ms:100s]

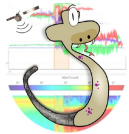
instant → slower

```
def filtered_B(start, stop):
    b = spz.get_data("cda/MMS1_FGM_SRVY_L2/mms1_fgm_b_gse_srvy_l2",
        start, stop)
    if b is None:
        return None
    return filtering.sosfiltfilt(sos, b)
```

```
def mms1_cos_theta_vb(start, stop):
    B,V=spz.get_data([
        spz.inventories.tree.cda.MMS.MMS1.FGM.MMS1_FGM_BRST_L2.mms1_fgm_b_gse_brst_l2_clean,
        spz.inventories.tree.cda.MMS.MMS1.DIS.MMS1_FPI_BRST_L2_DIS_MOMS.mms1_dis_bulkv_gse_brst
                  ], start, stop)

    B = B['Bx GSE', 'By GSE', 'Bz GSE']
    V = interpolate(B,V)
    cos_theta_vb = np.einsum('ij, ij->i', V, B)
            /(np.linalg.norm(B, axis=1)*np.linalg.norm(V, axis=1))

    return cos_theta_vb
```

The ecosystem

SciQLop
the IDE, browse, visualise, script, annotate, share



Plugins

AppStore

AstraLint
data quality, validation, standards compliance

pip install astralint

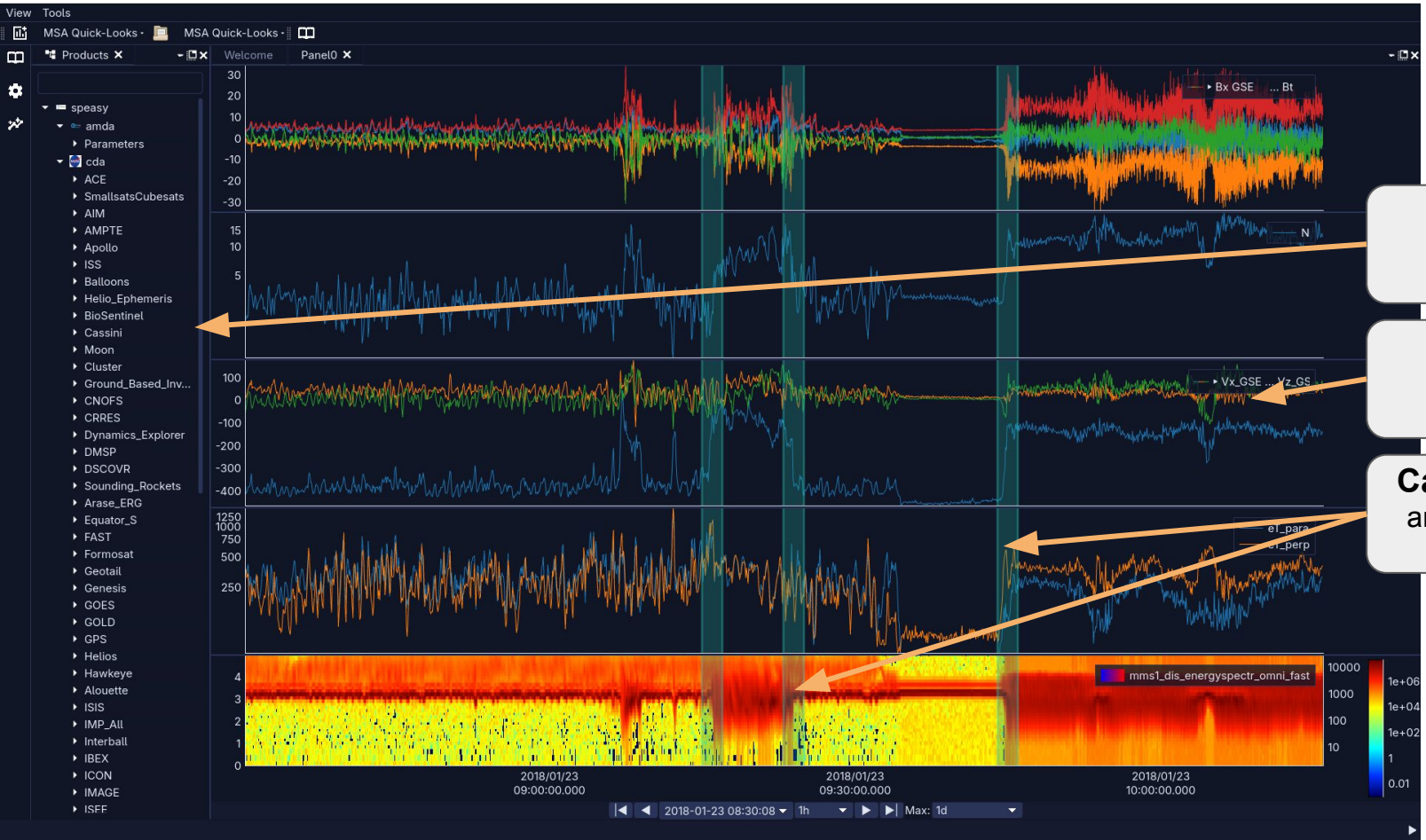
Speasy
unified access, AMDA, CDA, CSA, SSC…

pip install speasy

PyCDF++
super-fast C++ CDF reader

pip install pycdfpp

SciQLop: the **I**ntegrated **D**evelopment **E**nvironment (**IDE**)



Product tree
File explorer for datasets

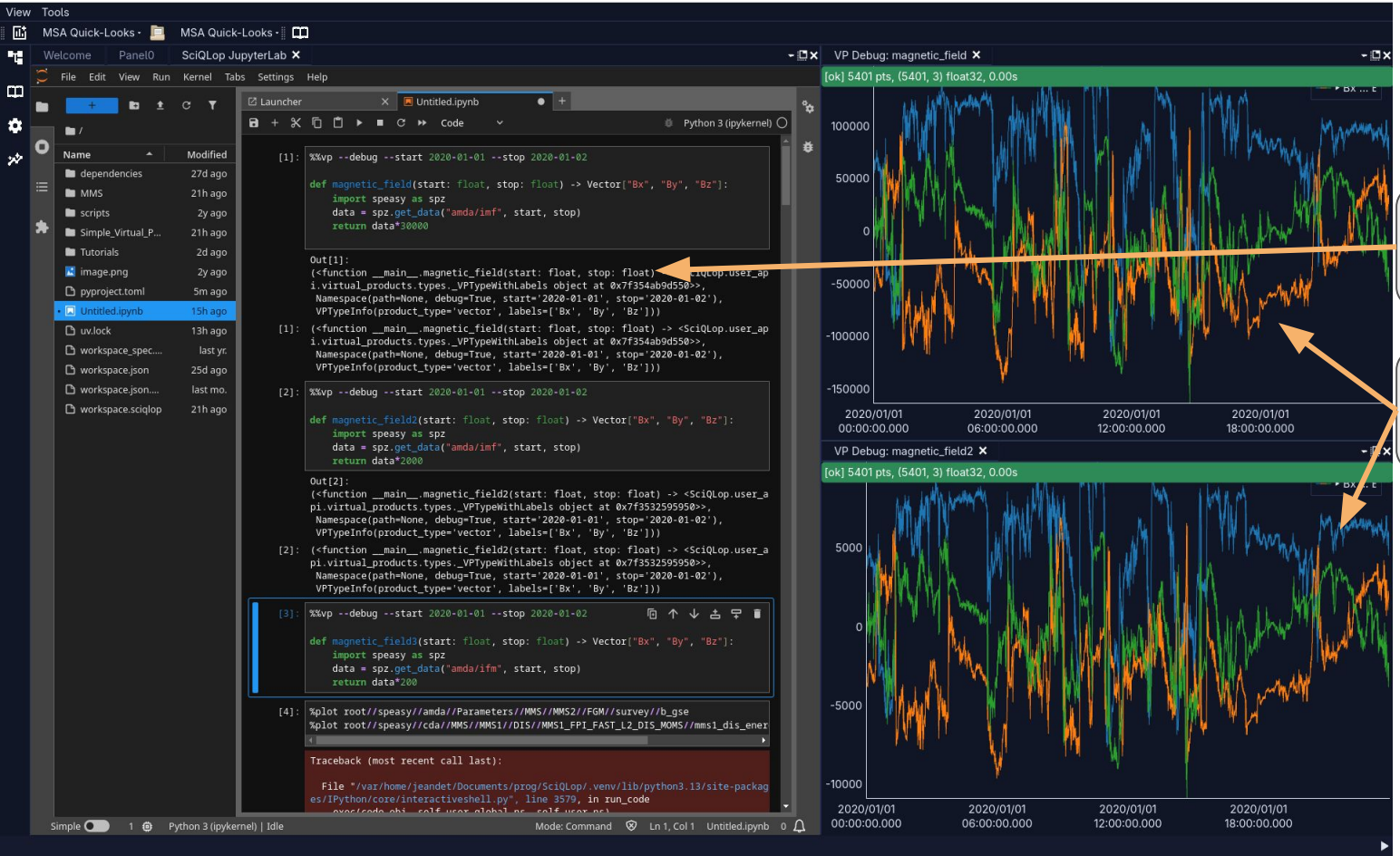
Plots panels

Catalogs overlay
annotate and share events

SciQLop: the **I**ntegrated **D**evelopment **E**nvironment (**IDE**)

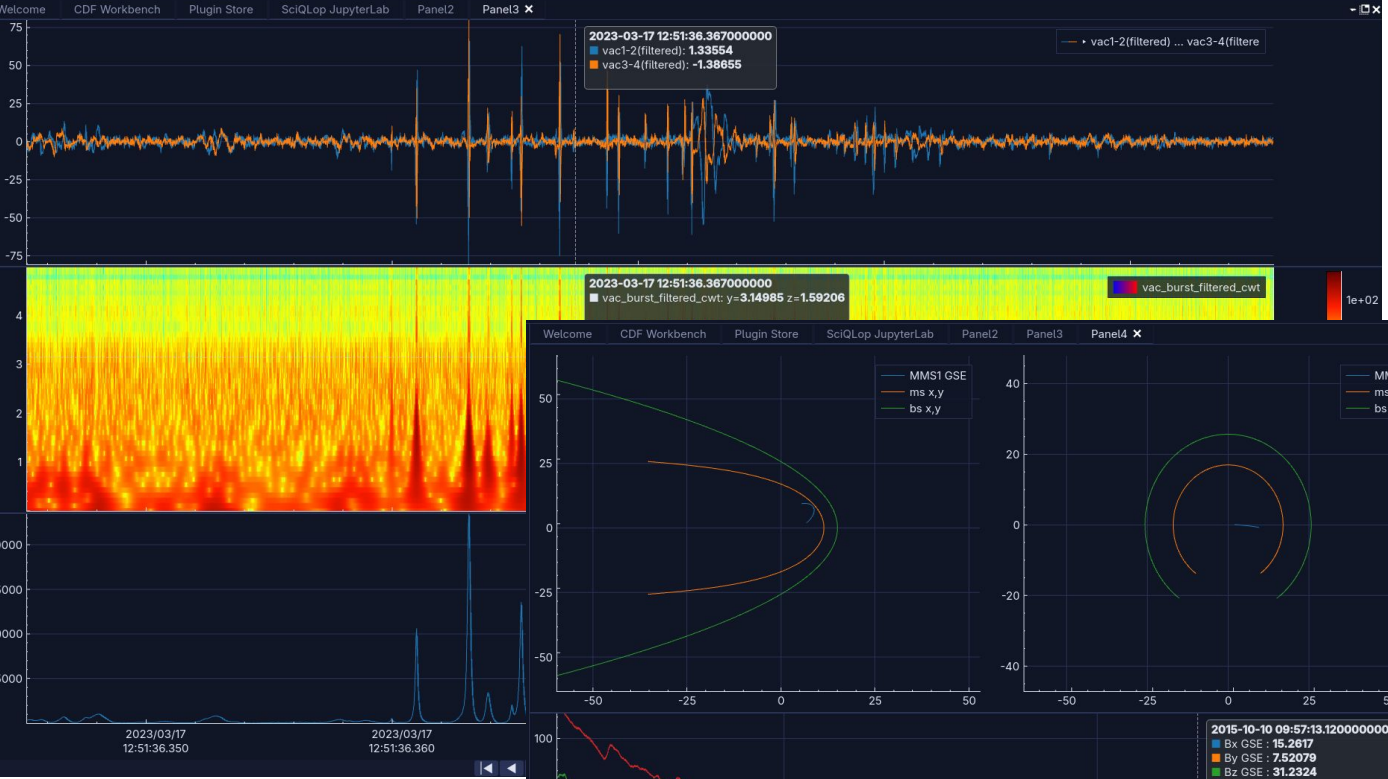
```
%%vp --path "mms/mirror"

def mirror_mode_threshold(start: float, stop: float) -> Scalar["Mirror mode threshold"]:
    mms1 = spz.inventories.data_tree.cda.MMS.MMS1
    products = [
        mms1.DIS.MMS1_FPI_FAST_L2_DIS_MOMS.mms1_dis_temppara_fast,
        mms1.DIS.MMS1_FPI_FAST_L2_DIS_MOMS.mms1_dis_tempperp_fast,
        mms1.FGM.MMS1_FGM_SRVY_L2.mms1_fgm_b_gse_srvy_l2,
        mms1.DIS.MMS1_FPI_FAST_L2_DIS_MOMS.mms1_dis_numberdensity_fast,
    ]
    tpara, tperp, b, n = spz.get_data(products, start, stop)
    if any(v is None for v in (tpara, tperp, b, n)):
        return None

    b = interpolate(tperp, b)
    Pperp = tperp * n * 1e6
    beta_perp = 2 * cst.mu_0 * cst.e * Pperp / (b["Bt"] * 1e-9) ** 2
    return beta_perp * 1000. * (tperp / tpara - 1)
```

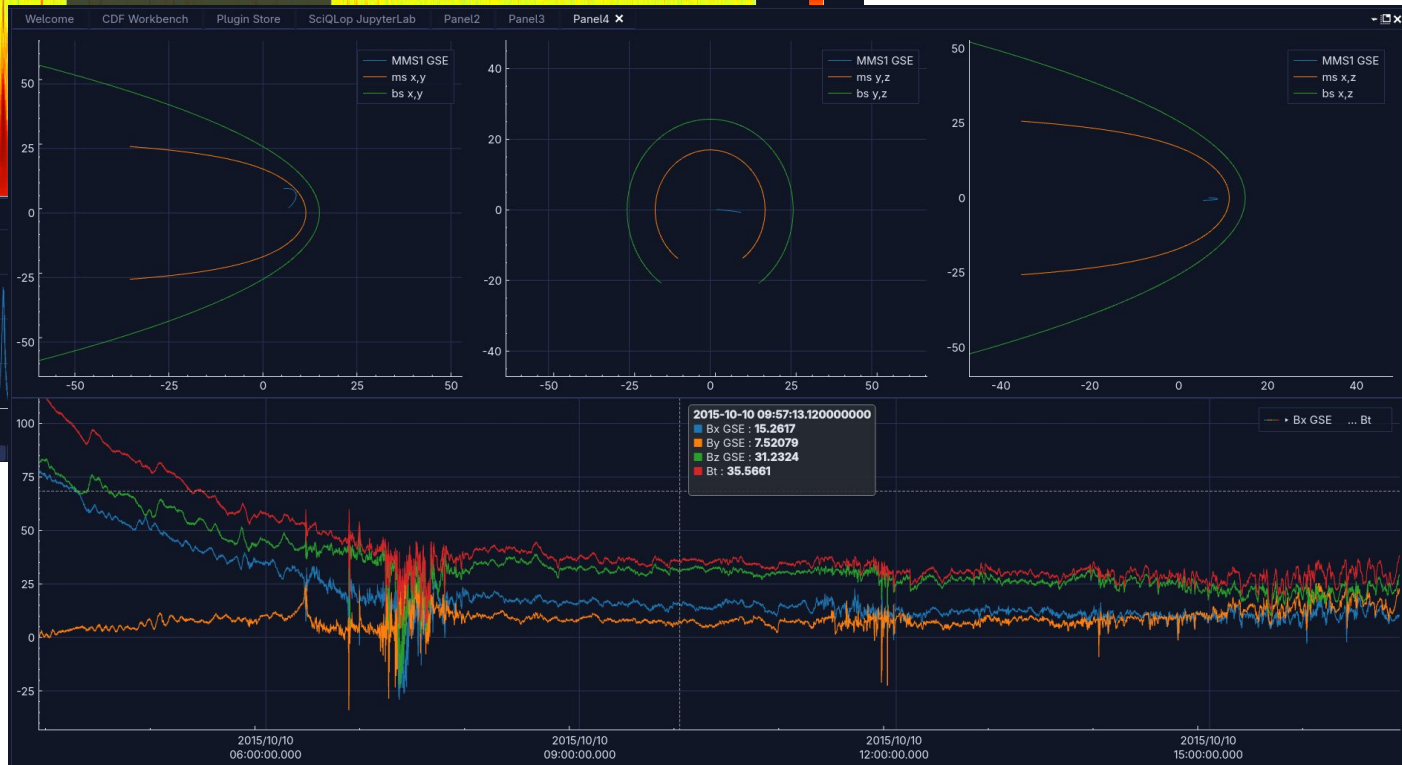
SciQLop: the **I**ntegrated **D**evelopment **E**nvironment (**IDE**)



JupyterLab
Write your own code

Debug plot
Quick debug feedback


Welcome
CDF Workbench
Plugin Store
SciQLop JupyterLab
Panel2
Panel3
2023-03-17 12:51:36.367000000
vac1-2(filtered): 1.33554
vac3-4(filtered): -1.38655
vac1-2(filtered) ... vac3-4(filtere
2023-03-17 12:51:36.367000000
vac_burst_filtered_cwt: y=3.14985 z=1.59206
vac_burst_filtered_cwt
1e+02
2023/03/17
12:51:36.350
2023/03/17
12:51:36.360


Welcome
CDF Workbench
Plugin Store
SciQLop JupyterLab
Panel2
Panel3
Panel4
MMS1 GSE
ms x,y
bs x,y
MMS1 GSE
ms y,z
bs y,z
MMS1 GSE
ms x,z
bs x,z
2015-10-10 09:57:13.120000000
Bx GSE : 15.2617
By GSE : 7.52079
Bz GSE : 31.2324
Bt : 35.5661
Bx GSE ... Bt
2015/10/10
06:00:00.000
2015/10/10
09:00:00.000
2015/10/10
12:00:00.000
2015/10/10
15:00:00.000

Plugins

- Calude, OpenCode, Github Copilot agents
  - Create/”see” plot panels
  - Find products
  - Create/modify Jupyter Notebooks
- CDF explorer
  - Plot any variable
- Radio astronomy
  - Curated list of space missions/instruments products
  - Sunpy/Radio Spectra integration
- Bepicolombo MSA
  - Custom panels

Conclusion

https://github.com/SciQLop

In v 0.12 now	build for your instrument	On the roadmap
• 65k+ products • Workspaces • JupyterLab • Built-in DSP functions • AI assistance • Plugins • AppStore • Catalogs (Local, Remote, *collaborative)*	• Plugins SDK • Custom data providers • Instrument specific panels • Custom pipelines • Share on AppStore	• Workspaces sharing • More UX features • More plugins • Collaborative JupyterLab/UI sessions • More data sources

open source · MIT/GPL license · supported by CDPP · contributions welcome

Workshop SciQLop, 15-17/09/2026, Jussieu (Paris)

- Talks
  - SciQLop success stories
  - SciQLop use cases
- Hands on