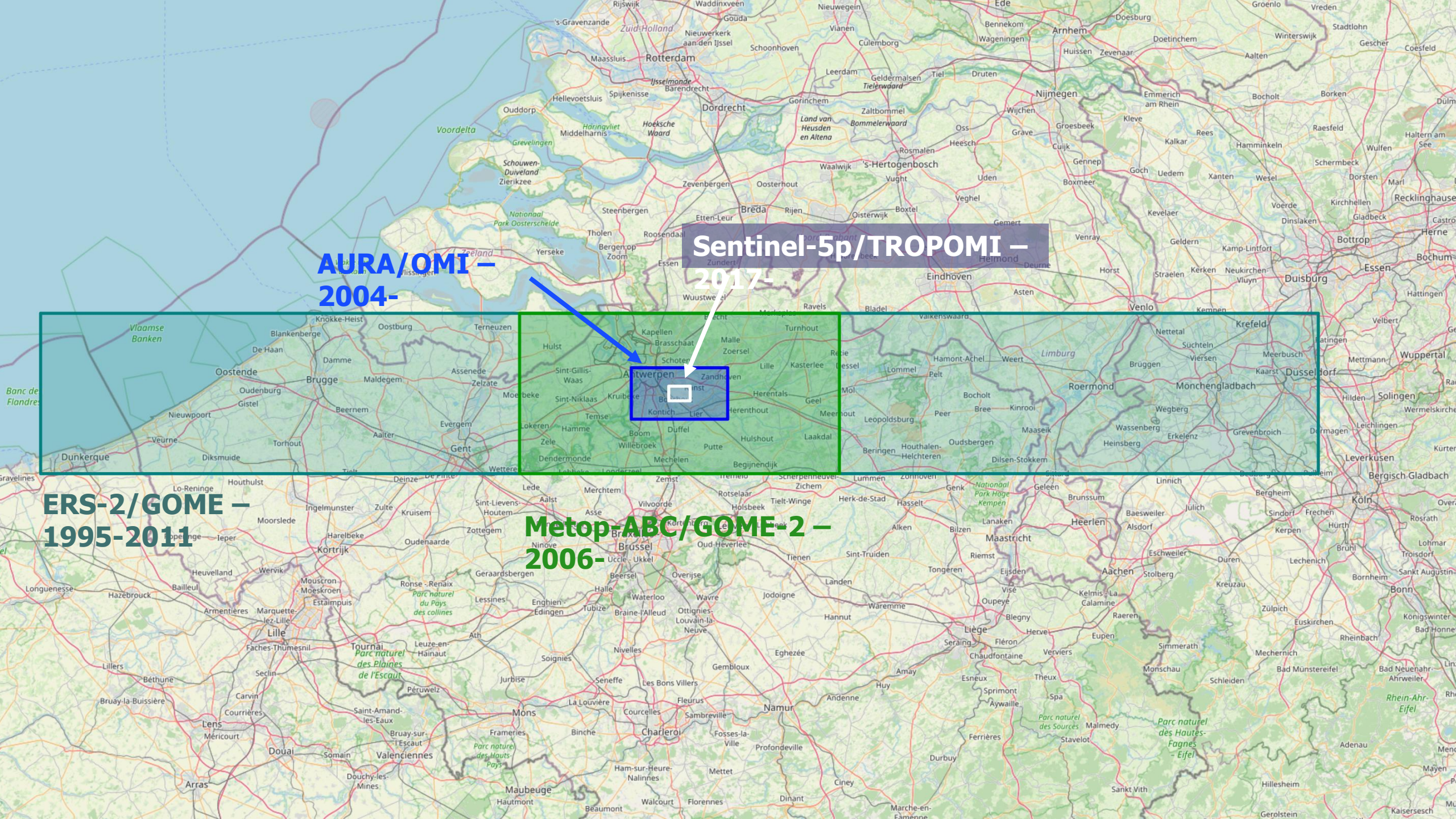
Caeloscope

Sentinel-5P atmospheric products for integration in Terrascope

Why Sentinel-5P?

STATE-OF-THE-ART FOR AQ

- Launch: October 2017.
- First Copernicus Sentinel dedicated to air quality, ozone, and climate.
- Relatively high spatial resolution of 3.5 × 5.5 km^2.


AURA/OMI – 2004-
Sentinel-5p/TROPOMI – 2017-
ERS-2/GOME – 1995-2011
Metop-ABC/GOME-2 – 2006-

How it began...

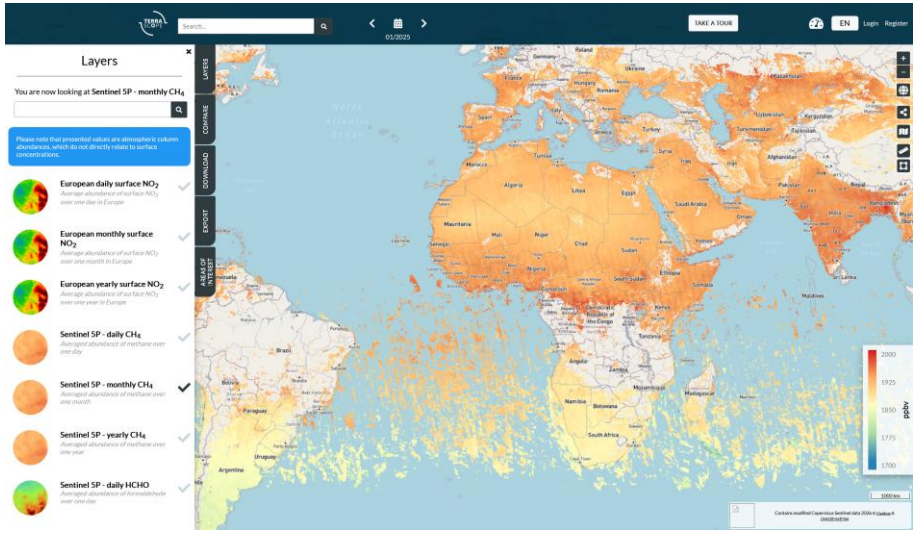
The Belgian Institute for Space Aeronomy has a strong involvement in S-5P/TROPOMI activities.

- PI for prototype developments and maintenance of S5P SO_2 and H2CO data products, and Co-I for total ozone developments.
- Co-I for L2 Algorithm Evolution in ESA's Mission Performance Center (MPC), as Level-2 Expert Support Laboratory (ESL-L2).
- Lead of the MPC-Validation Data Analysis Facility (VDAF)
- PI of "Sentinel-5 Precursor Level 2 Processor Component Development – BIRA and RAL contributions", ESA Contract No. 4000107744/13/NL/IB/lf (2013-2018).
- => Series of ESA PRODEX contracts (TRACE-S5P)
  First Terrascope-related activities proposed for TRACE-S5P IV (PEA 400010559): integration of NO_2 and CO L3. Release in 2021.

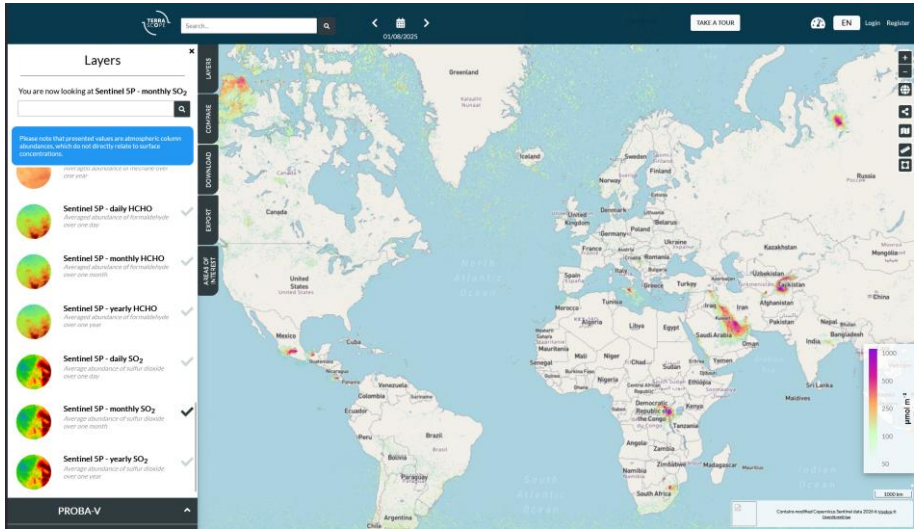
What's already there…

- Daily, monthly, and yearly averages of:

  CO, CH_4, HCHO, NO_2, SO_2



Average methane mixing ratio for the year 2025.

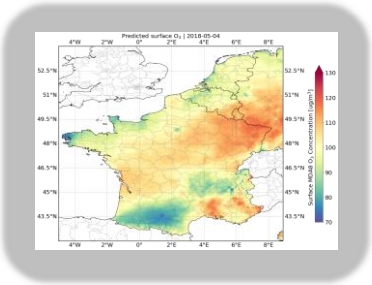


Average sulphur dioxide column for August 2025.

- Several products provide vertical column data
  - ❑ Useful to seasoned EO community
  - ❑ Broader society needs derived parameters, like surface concentration.

CAELOSCOPE: What will come…

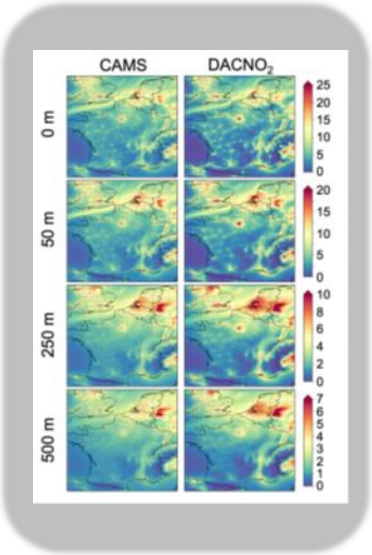
DERIVED PRODUCTS: A STEP CLOSER TOWARDS THE USER COMMUNITY



Surface O_3

- 2×2 km^2 Ozone surface concentration.
  from machine learning algorithm.

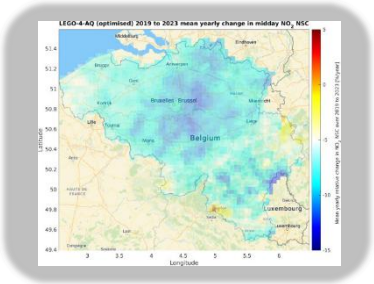
AQ, Health authorities



3D NO_2

- 2×2 km^2 3D NO_2 concentration [0m – 5000m]
  from machine learning algorithm.

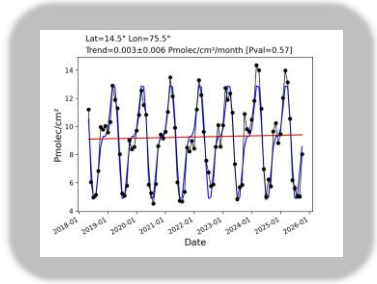
Scientific community: high-res NO_2 profiles.



Synergetic satellite / *in-situ* surface NO_2

- 1×1 km^2 seasonal + yearly NO_2 surface concentration (Belgium)
  from TROPOMI + *in situ* observational data.

Policy, health, local authorities.



- Trace gas trend data and methodology.

AQ, climate, health, ….

CAELOSCOPE: What will come…

DERIVED PRODUCTS: A STEP CLOSER TOWARDS THE USER COMMUNITY

Guiding the user

1. Think of non-expert users

2. Gradually increase detail:
   From simple README file,
   via Notebook examples,
   to in-depth information: ATBD, papers

Using the data

A JUPYTER NOTEBOOK STORY.

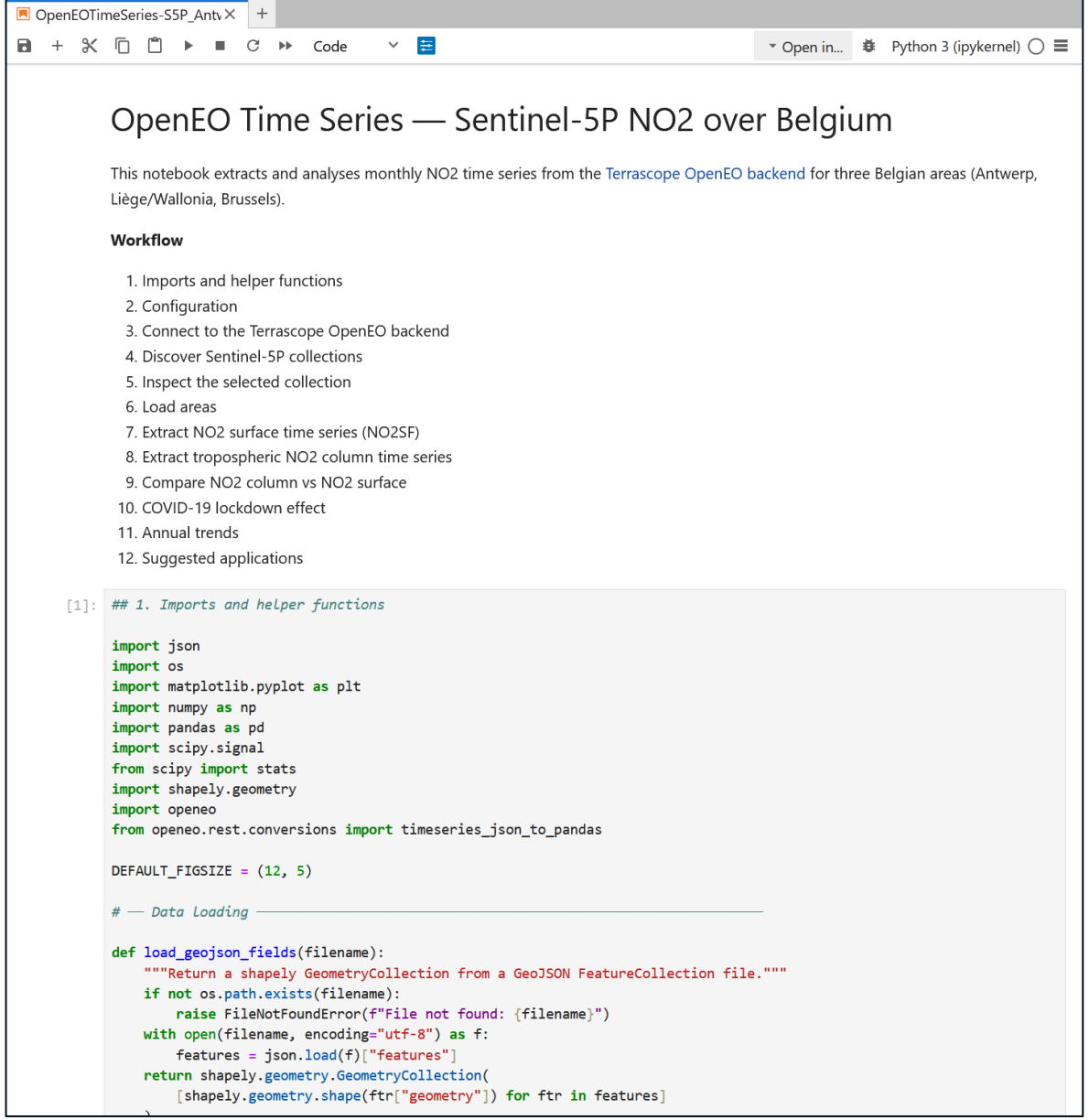
Options:

1. Do your analysis at notebooks.terrascope.be

2. Run in your own Jupyter environment and use openEO API to interact with Terrascope.

Story on Nitrogen Dioxide (NO_2):

- Discover the different NO_2 datasets.
- Define your regions of interest.
- Ingest the data as timeseries.
- Discover events and trends for the areas.



Thank you, Isabelle De Smedt

Using the data

A JUPYTER NOTEBOOK STORY.

1. Authenticate first (obtain token).
2. Find the S-5P datasets in the catalogue.

3. Connect to the Terrascope OpenEO backend

Authenticates using EGI OIDC. A browser pop-up or link appears on first login; subsequent runs use a stored refresh token.

[3]:
```
con = openeo.connect("https://openeo.vito.be").authenticate_oidc(provider_id="egi")
print(con)
```

```
Authenticated using refresh token.
<Connection to 'https://openeo.vito.be/openeo/1.2/' with OidcBearerAuth>
```

4. Discover Sentinel-5P collections

Filter the full collection list to Terrascope S5P datasets, excluding deprecated `V1` entries.

[4]:
```
collections = list(con.list_collection_ids())
s5p = [c for c in collections if "S5P" in c and "V1" not in c]
print(f"Available Terrascope Sentinel-5P collections ({len(s5p)}):")
for c in s5p:
    print(" ", c)
```

```
Available Terrascope Sentinel-5P collections (27):
  TERRASCOPE_S5P_L3_CH4_TD
  TERRASCOPE_S5P_L3_CH4_TM
  TERRASCOPE_S5P_L3_CH4_TY
  TERRASCOPE_S5P_L3_CO_TD
  TERRASCOPE_S5P_L3_CO_TM
  TERRASCOPE_S5P_L3_CO_TY
  TERRASCOPE_S5P_L3_HCHO_TD
  TERRASCOPE_S5P_L3_HCHO_TM
  TERRASCOPE_S5P_L3_HCHO_TY
  TERRASCOPE_S5P_L3_NO2_TD
  TERRASCOPE_S5P_L3_NO2_TM
  TERRASCOPE_S5P_L3_NO2_TY
  TERRASCOPE_S5P_L3_SO2CBR_TD
  TERRASCOPE_S5P_L3_SO2CBR_TM
  TERRASCOPE_S5P_L3_SO2CBR_TY
  TERRASCOPE_S5P_L3_NO2EU_TD
  TERRASCOPE_S5P_L3_NO2SF_TD
  TERRASCOPE_S5P_L3_NO2EU_TM
  TERRASCOPE_S5P_L3_NO2SF_TM
  TERRASCOPE_S5P_L3_NO2EU_TY
  TERRASCOPE_S5P_L3_NO2SF_TY
  TERRASCOPE_S5P_L3_NO2_CAMS_TD
  TERRASCOPE_S5P_L3_NO2_CAMS_TM
  TERRASCOPE_S5P_L3_NO2_CAMS_TY
  TERRASCOPE_S5P_L3_NO2_SURFACE_TD
  TERRASCOPE_S5P_L3_NO2_SURFACE_TM
  TERRASCOPE_S5P_L3_NO2_SURFACE_TY
```

Using the data

A JUPYTER NOTEBOOK STORY

Define your areas of interest.

6. Load areas

Load the three Belgian polygons from the GeoJSON file.

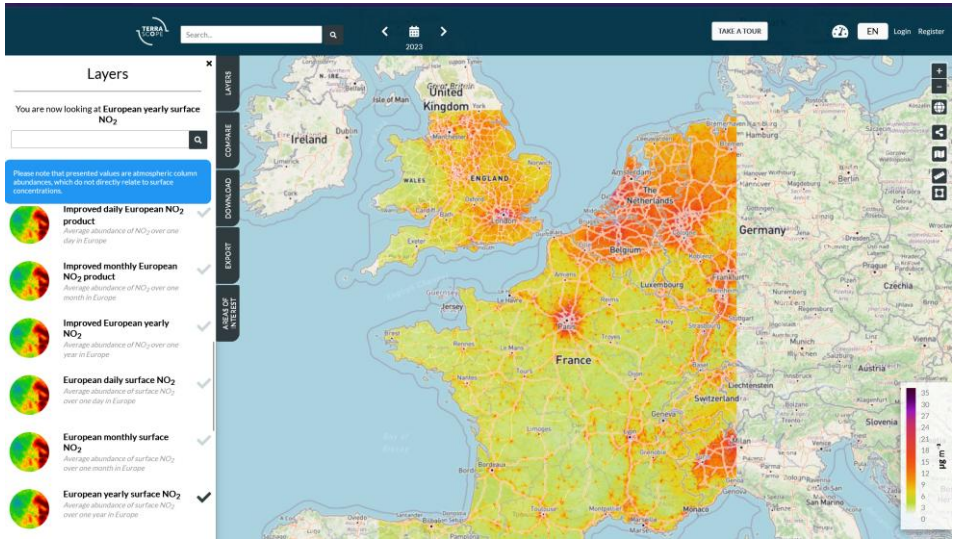
[16]:
```
import folium
from shapely.geometry import mapping
# ── Areas ──────────────────────────────────────────────────────────

GEOJSON_FILE = "areas_Belgium.geojson"
AREA_NAMES   = {0: "Antwerp", 1: "Namur", 2: "Brussels"}

fields = load_geojson_fields(GEOJSON_FILE)
print(f"Loaded {len(fields.geoms)} area(s): {list(AREA_NAMES.values())}")
#fields

# Create a map centered on your geometries
centroid = fields.centroid
m = folium.Map(location=[centroid.y, centroid.x], zoom_start=7)

# Add each geometry as a layer
for name, geom in zip(AREA_NAMES.values(), fields.geoms):
    folium.GeoJson(
        mapping(geom),
        name=name,
        tooltip=name,
        style_function=lambda _: {
            "fillColor": "#3388ff",
            "color": "#0000ff",
            "weight": 2,
            "fillOpacity": 0.2,
        }
    ).add_to(m)

folium.LayerControl().add_to(m)
m  # Renders inline in Jupyter
```

```
Loaded 3 area(s): ['Antwerp', 'Namur', 'Brussels']
```

[16]:

Using the data

A JUPYTER NOTEBOOK STORY



Average NO_2 surface concentration for 2024

7. Extract NO2 surface time series (NO2SF)

Loads `TERRASCOPE_S5P_L3_NO2SF_TM`, aggregates spatially (mean) over the three areas, and downloads the result as JSON. Re-run this cell only when the parameters in section 2 change.

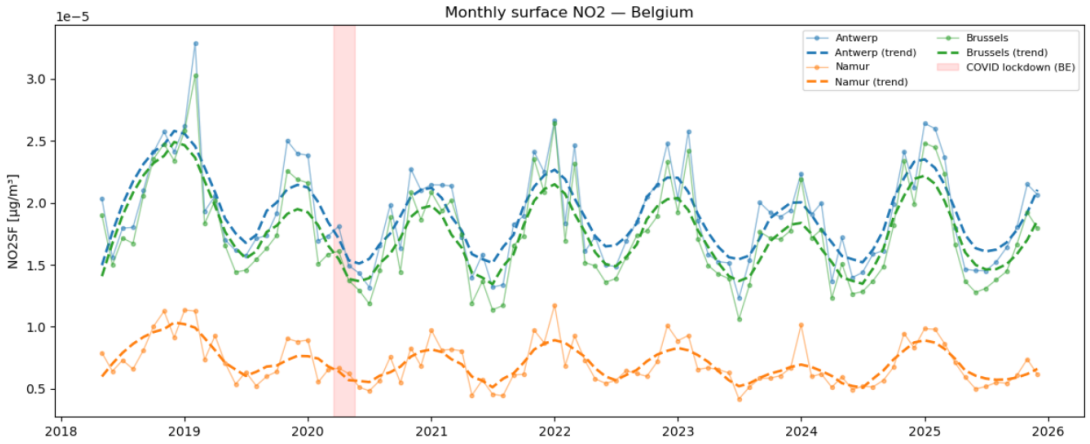
[7]:
```
file_no2sf = extract_timeseries(con, COLLECTION_SF, BAND_SF, BBOX, DATES, fields)
ts_no2sf   = load_timeseries(file_no2sf, scale=SCALE_SF, area_names=AREA_NAMES)
ts_no2sf.head()
```

```
Downloaded → timeseries_TERRASCOPE_S5P_L3_NO2SF_TM_NO2SF.json
```

[7]:

date	Antwerp	Liège/Wallonia	Brussels
2018-05-01	0.000020	0.000008	0.000019
2018-06-01	0.000016	0.000006	0.000015
2018-07-01	0.000018	0.000007	0.000017
2018-08-01	0.000018	0.000007	0.000017
2018-09-01	0.000021	0.000008	0.000021

[30]:
```
plot_timeseries(ts_no2sf, title="Monthly surface NO2 — Belgium", ylabel="NO2SF [µg/m³]")
```



A spike is visible in the Antwerp area around April 2019. Cross-checking on the Terrascope viewer confirms the anomaly.

8. Extract tropospheric NO2 column time series

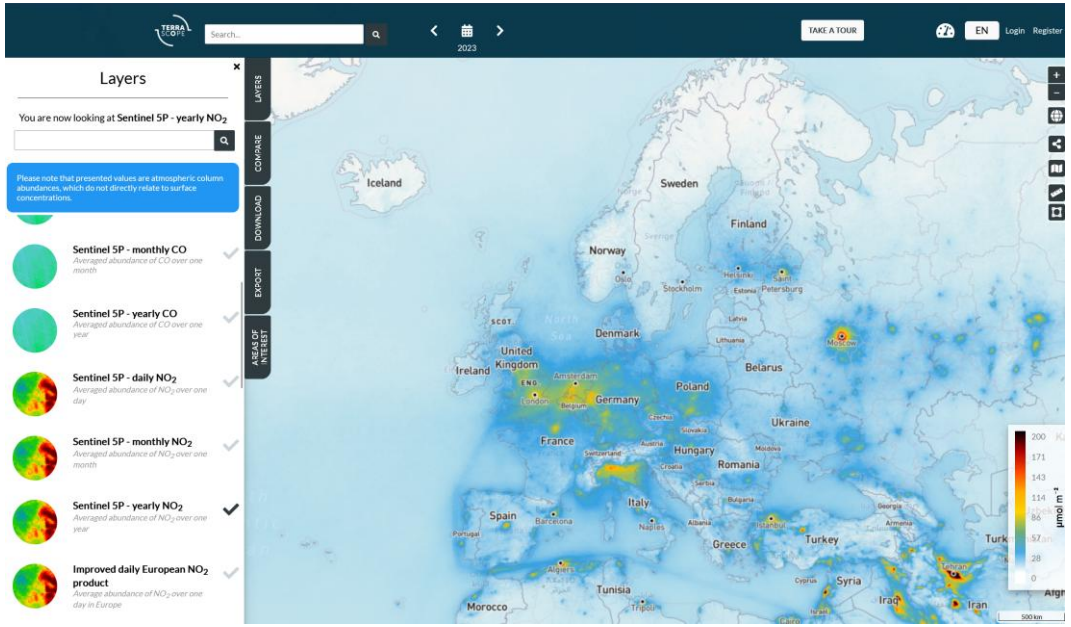
Uses `TERRASCOPE_S5P_L3_NO2_TM` — the total tropospheric column density without surface correction. Compare with NO2SF in section 9.

[ ]:
```
file_no2 = extract_timeseries(con, COLLECTION_NO2, BAND_NO2, BBOX, DATES, fields)
```

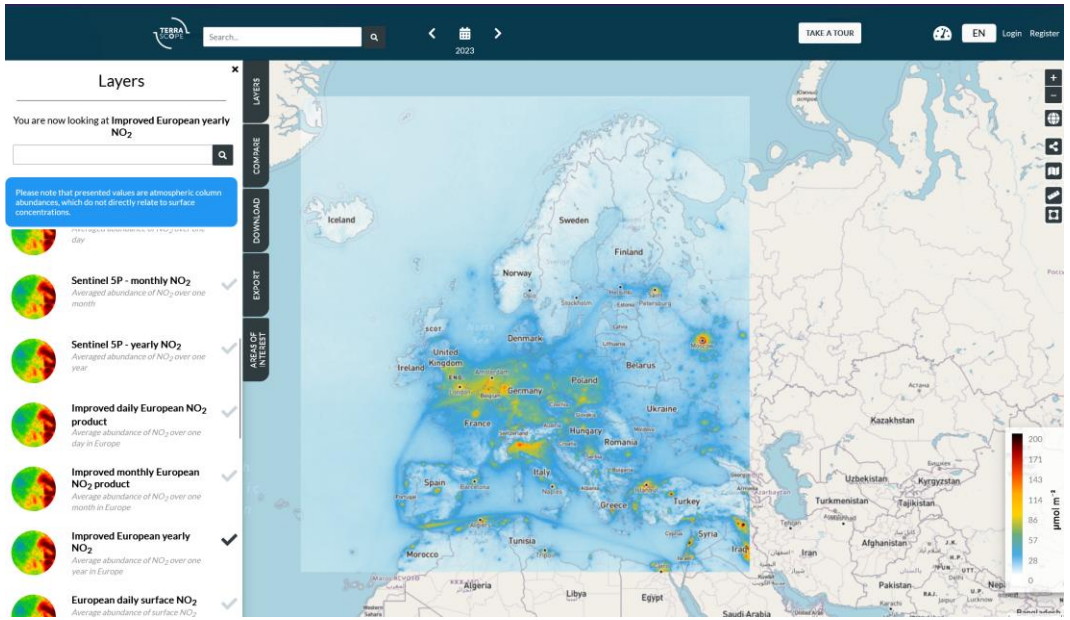
Using the data

A JUPYTER NOTEBOOK STORY

Ingest 2 topospheric column NO_2 data sets.



Operational tropospheric NO_2 for 2023 (Copernicus)



'Improved' tropospheric NO_2 for 2023 (CAMS profile, KNMI)

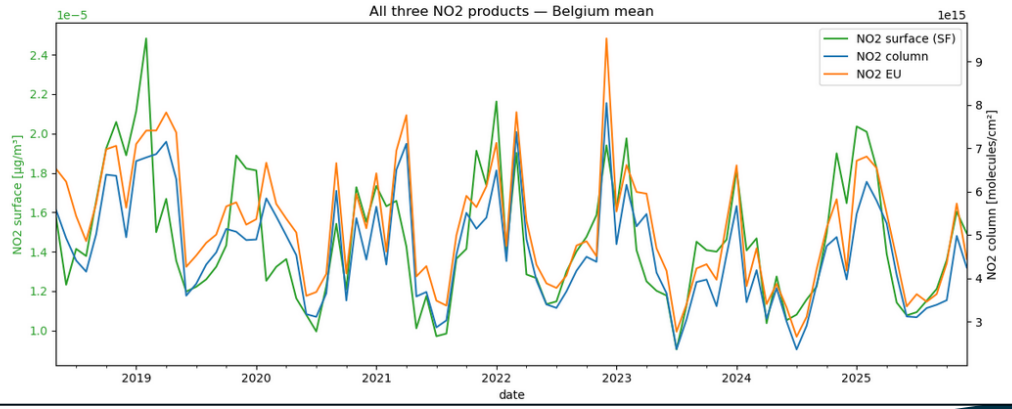
Using the data

A JUPYTER NOTEBOOK STORY

10. Compare all three NO2 products

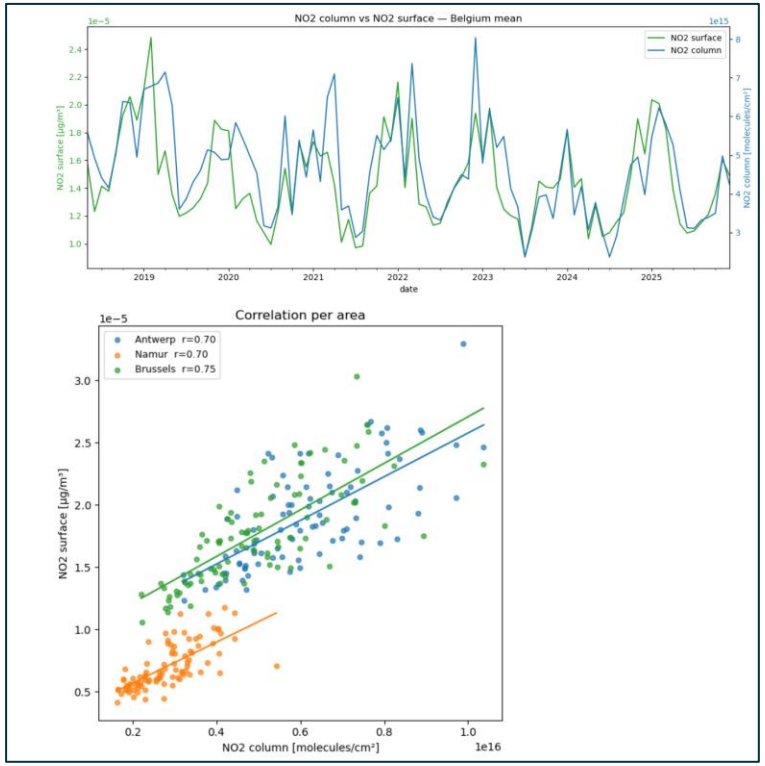
Overlay the Belgium mean for all three products, then show pairwise correlations against the surface NO2 reference (NO2SF).

[41]:
```
# Overlay: Belgium mean — dual y-axis (surface left, column right)
fig, ax1 = plt.subplots(figsize=DEFAULT_FIGSIZE)
ax2 = ax1.twinx()

color_sf = "tab:green"
ts_no2sf.mean(axis=1).plot(ax=ax1, label="NO2 surface (SF)", color=color_sf)
ax1.set_ylabel("NO2 surface [µg/m³]", color=color_sf)
ax1.tick_params(axis="y", labelcolor=color_sf)

colors_col = ["tab:blue", "tab:orange"]
for (ts, label), color in zip([(ts_no2, "NO2 column"), (ts_no2eu, "NO2 EU")], colors_col):
    ts.mean(axis=1).plot(ax=ax2, label=label, color=color)
ax2.set_ylabel("NO2 column [molecules/cm²]", color="black")

# Combined legend
lines1, labels1 = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax1.legend(lines1 + lines2, labels1 + labels2, loc="upper right")
ax1.set_title("All three NO2 products — Belgium mean")
plt.tight_layout()
plt.show()

# Pairwise correlations vs NO2SF (used as reference)
plot_correl(ts_no2,   ts_no2sf, label1="NO2 column", label2="NO2 surface")
plot_correl(ts_no2eu, ts_no2sf, label1="NO2 EU",     label2="NO2 surface")
```

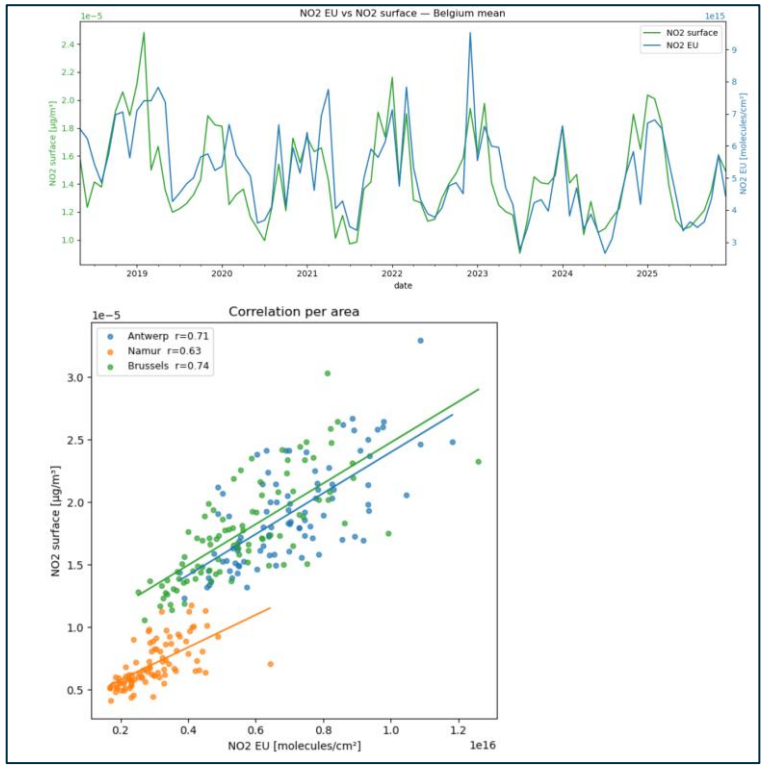


Using the data

A JUPYTER NOTEBOOK STORY



NO2 surface vs. NO2 operational



NO2 surface vs. NO2 EU

Using the data

A JUPYTER NOTEBOOK STORY

COVID-19
Belgian lockdown 18 March to 18 May 2020.

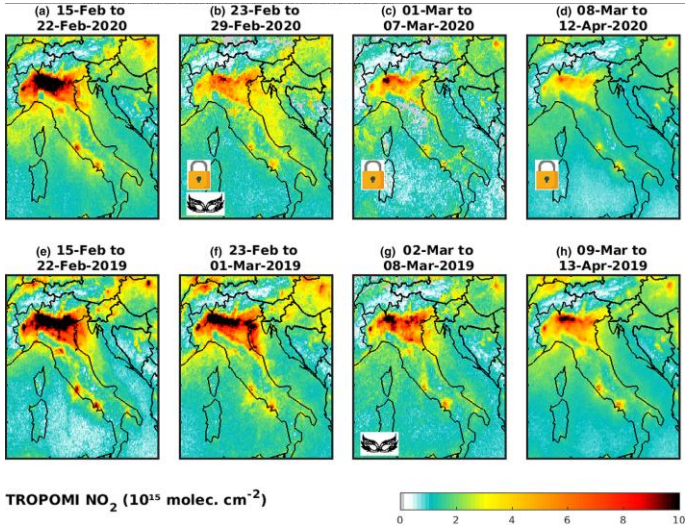
11. COVID-19 lockdown effect

Belgium's national lockdown ran from **18 March to 18 May 2020**. Surface NO2 is a direct proxy for traffic and combustion emissions, making this dataset ideal for quantifying the effect.

[42]:
```
# Compare Mar–May 2020 (lockdown) vs Mar–May 2019 (reference year)
spring = [3, 4, 5]
ref  = ts_no2sf[(ts_no2sf.index.year == 2019) & (ts_no2sf.index.month.isin(spring))]
lock = ts_no2sf[(ts_no2sf.index.year == 2020) & (ts_no2sf.index.month.isin(spring))]

reduction_pct = (lock.mean() - ref.mean()) / ref.mean() * 100
print("NO2SF reduction during lockdown (Mar–May 2020 vs 2019):")
print(reduction_pct.round(1).to_frame("Δ [%]"))
```

```
NO2SF reduction during lockdown (Mar–May 2020 vs 2019):
           Δ [%]
Antwerp    -11.4
Namur      -18.1
Brussels   -17.0
```

Nice use case for the Viewer!



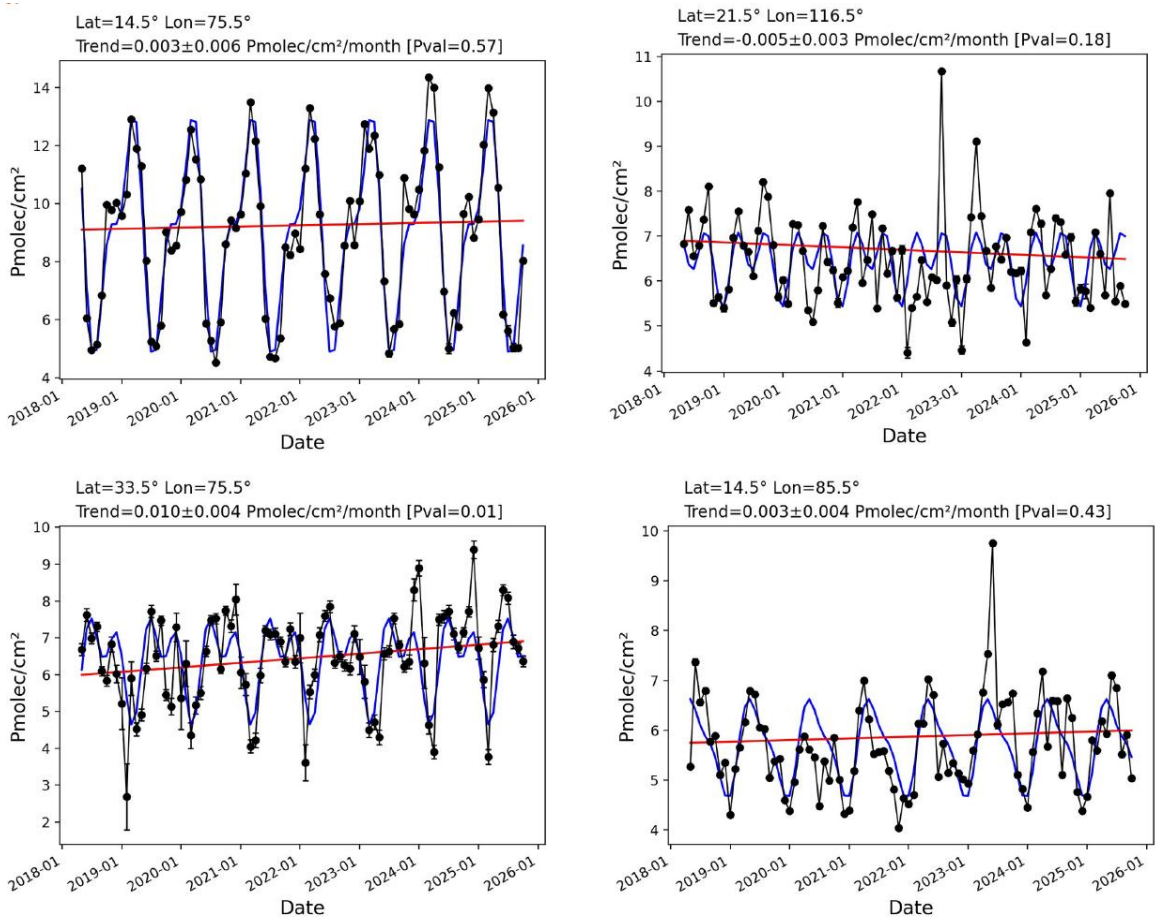
From: Bauwens *et al.*, 2020.

Using the data

A JUPYTER NOTEBOOK STORY

Trends

Note: So far for formaldehyde



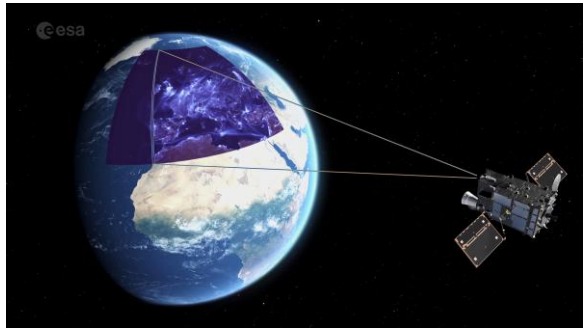
Formaldehyde local trends with P-values

Outlook

PRODUCT INPLEMENTATION

- **"Now":** 3D NO_2 concentration
- **September:** S-5P/in-situ NO_2
- **September-December:** Trends
- **December:** Ozone surface concentration.

2027: User promotion and interaction



Sentinel-5 (LEO, morning overpass)

Sentinel-4 (GEO over Europe, North Africa, hourly data)

Both still in commisionning.