SHERPA

Building a localized version of SHERPA

P. Thunis, E. Pisoni, D. De Marchi, A. De Meij, S. Zauli, A. Rey Pommier, R. Vignotto

SHERPA
Screening for High Emission Reduction Potential on Air

European Commission
Joint Research Centre

Software developed by TerrAria under the Contract Procedure no. JRC/IPR/2014/H.2/0023/NC

Possible uses of SHERPA

- You can use SHERPA either with
  - GUI, and default SRR (easier)
  - Python code, based on default SRR (more flexible)
  - Python code, to train your own SRR

SHERPA using python code and EU data

Python code based on default SRR

- https://github.com/enricopisoni/SHERPA-simulation
- All code and data to run the ‘scenario analysis’ are now available on the repository.

Python-code: SRR based on:
- 2022 CAMS v8 emissions including condensables
- 2021 IFS meteo
- EMEP v4.45
- 0.1x0.05 deg

Please contact us if you want to follow this path

Python code based on default SRR

Run module 1

After setting up the environment, you can then run module 1 through the following steps (please note that here below we included some additional python commands, mainly useful to show the content of the different input data required by module 1).

In [ ]:
```
#import required libraries
import xarray as xr
import pandas as pd
import os
import matplotlib.pyplot as plt
from module1 import module1
```

In [ ]:
```
#load basecase emissions
path_emission_cdf_test = './input/BC_emiss/BCemis_emepV434_camsV42_01_005_SecEmis.nc'
emis = xr.open_dataset(path_emission_cdf_test)
emis
```

In [ ]:
```
#load basecase concentrations
path_base_conc_cdf_test = './input/BC_concs/BCconc_emepV434_camsv42_01005_SURF_ug_PM25_rh50.nc'
conc = xr.open_dataset(path_base_conc_cdf_test)
conc
```

In [ ]:
```
#load the SRR model
path_model_cdf_test = './input/SRR/SR_SURF_ug_PM25_rh50.nc'
srr = xr.open_dataset(path_model_cdf_test)
srr
```

https://github.com/enricopisoni/SHERPA-simulation/blob/master/how_to_run_module1.ipynb

Python code based on default SRR

There are modules for:

- Scenario analysis
- Sectoral and precursor-based SA
- Atlas mode
- Health evaluation

SHERPA using python code and your own data

Python code, to train your own SRR

- https://github.com/enricopisoni/SHERPA-training

For this, you need:

- Your emission inventory (any spatial resolution) and meteorology
- A CTM to be run for around 10 simulations (training and validation to be defined)
- A «polygons definition» (i.e. Urban Audit)
- WGS84 is the preferred option (but can be adapted)

As an example at EU scale

ID	EMISSION INVENTORY	SHORT DESCRIPTION	GEOGRAPHICAL REDUCTIONS	SECTORAL REDUCTIONS
sce0	CAMS v8 (2022)	Basecase		
sce1	CAMS v8 (2022)	50%reduction NOx	all domain	all sectors
sce2	CAMS v8 (2022)	50%reduction VOC	all domain	all sectors
sce3	CAMS v8 (2022)	50%reduction NH3	all domain	all sectors
sce4	CAMS v8 (2022)	50%reduction PPM	all domain	all sectors
sce5	CAMS v8 (2022)	50%reduction SO2	all domain	all sectors
sce6	CAMS v8 (2022)	50%reduction allPoll	all domain	all sectors
sce7	CAMS v8 (2022)	25%reduction	combined reduction (areas around Madrid, Milan, Paris, Berlin, Warsaw, Athens)	residential and transport sectors

Procedure to build your own SRR

- Choose a reference year for your runs
- Run the CTM on a number of training and validation scenario
- Train the SRR ... done in 2 steps:
  - Identification of «omega» ... Define how the emissions (surrounding the receptor cell) influence the concentrations at the receptor point
    - «Maximum area of influence»
    - «moving window» size
  - Identification of «alpha» ... Define relative importance of the different precursors
- Validation ... Comparing AQM and SHERPA:
  - Based on scatter plots, bias percentage map

Procedure to build your own SRR

The procedure is not fully codified (but by end of 2026 we should have a more guided way to build your own SHERPA)

If you want to start before end of 2026, please contact us to work together on this

Any question?