The Python Environment for Reaction Mechanism/Mathematics (PERM)

Barron Henderson[1,2], William Vizuete[1], Harvey Jeffries[1]

[1]*Dept. of Environmental Science and Engineering, UNC Chapel Hill* [2]*Atmospheric Modeling and Analysis Division, U.S. EPA*

MOTIVATION

Mitigating pollutant formation, global climate change, and toxic exposures require understanding how concentrations form. The formation of pollutant concentrations involves many competing processes and reactions. To capture details of the concentration development, most models include Process Analysis (PA) capabilities. The PA capabilities, however, are difficult to use because PA data is large and can be difficult to use. That's where the Python Environment for Reaction Mechanisms/Mathematics (PERM) comes in. PERM is a suite of tools designed to make interaction with PA data easier, and maybe even fun. Improving *ease of use* will allow researchers and regulators to interact more fully with their data. Improved interaction with the data allows better understanding and, potentially, more effective policy.

PREPARING PA DATA FOR PERM

There are three steps required for using PERM.

1. determine the goal of your evaluation
2. run your model with PA enabled
3. extract the PA data for you evaluation

The first step will obviously influence the other two and, unfortunately, you're on your own. Enabling PA is specific to the model you're using, but PERM has been used with PA from CMAQ, CAMx, DSMACC, Morpho, and WRF-Chem. When using a 3-D model, the data is most useful when extracted for a particular analysis volume. More information on constructing an analysis volume is available at the pyPA website (https://dawes.sph.unc.edu/trac/pyPA).

BASIC PERM OBJECTS

PERM creates an environment for interaction with a mechanism. The mechanism has 2 fundamental objects: species and reactions. A species can be either a single species or a species family. For instance, NOx is the sum of NO and NO2 (syntax: NOx = NO + NO2). Similarly, a reaction can be either a single reaction or a net reaction. For instance, net PAN formation is the sum of production, thermal decomposition, and photolysis (syntax: net_pan = IRR_88 + IRR_89 + IRR_90). The mechanism objects are augmented by modeled physical processes (i.e. emissions, advection, etc). These objects provide the foundation for all of PERM's functionality.

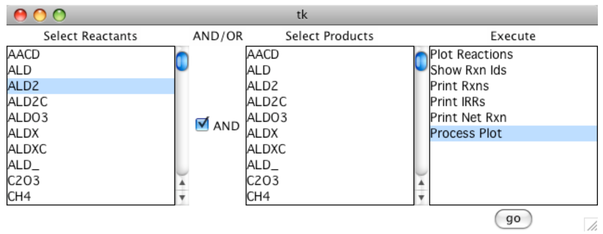
USING THE PERM ENVIRONMENT

PERM is a portal to Process Analysis data and a repository of chemical mechanisms. PERM stores representations of common chemical models that can be expanded by the user. Each chemical mechanism has a listing of reactions, species, species groups. The species can be used to query the mechanism for reactions. The query is the basis for much of PERM's utility. The query can be used to print, plot, or apply mathematics to reaction production/consumption of species. PERM offers three ways to do anything: a graphical user interface, a interactive environment, and scripting.

PROCESS PLOT EXAMPLE

Graphical User Interface

```
$ python -m perm -g -c cb05_cmaq test.mrg.nc
```



Interactive Environment

```
$ python -m perm -i -c cb05_cmaq test.mrg.nc

>>> plot_proc(ALD2)
```

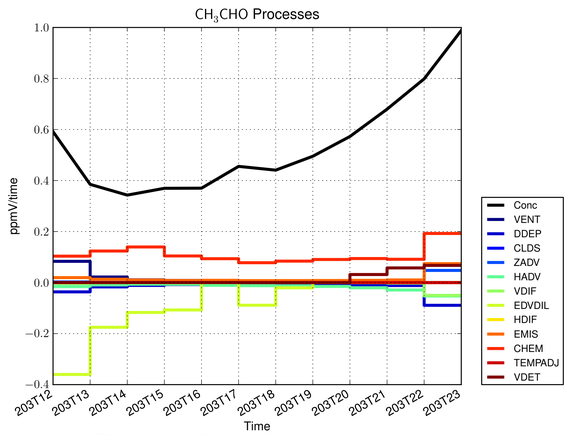
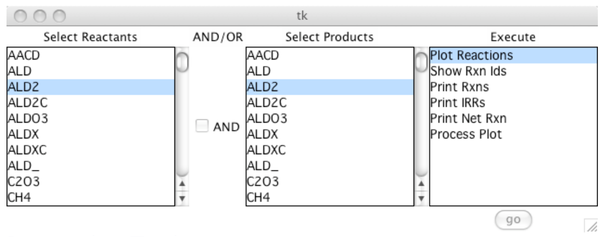


Figure 1: Process plot standard figure.

REACTION PLOT EXAMPLE

Graphical User Interface



Interactive Environment

```
$ python -m perm -i -c cb05_cmaq test.mrg.nc

>>> plot_rxns(ALD2, ALD2, logical_and = False)
```

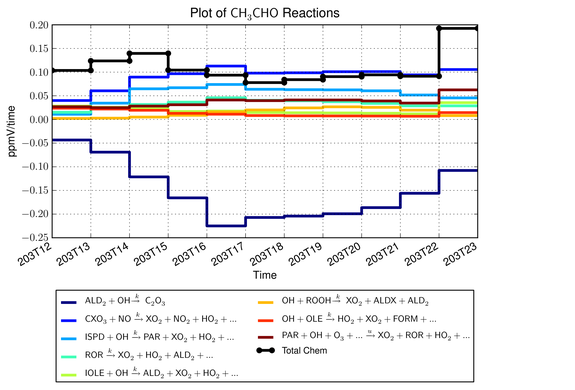


Figure 2: Standard reaction plot output.

SILLMAN NOX/VOC-SENSITIVITY INDICATOR EXAMPLE

Scripted Operation

```
$ python -m perm -c cb05_cmaq test.mrg.nc sillman.py
```

Listing 1: sillman.py

```
PEROXIDES = ROOH + H2O2
nrxn_prod_peroxides = make_net_rxn(products = \
  PEROXIDES)
nrxn_prod_hno3 = make_net_rxn(products = HNO3)
prooh = nrxn_prod_peroxides[PEROXIDES]
phno3 = nrxn_prod_hno3[HNO3]
sillman_ratio = prooh/phno3
daily_sillman_ratio = prooh.sum()/phno3.sum()
plot(sillman_ratio, path = 'sillman_ratio.pdf', \
  axis_settings = dict(title='Sillman Ratio \
  (Integrated: %.2f)' % daily_sillman_ratio \
  ,ylabel=r'$P$(ROOH):$P$(HNO3)',xlabel = 'Time'))
```

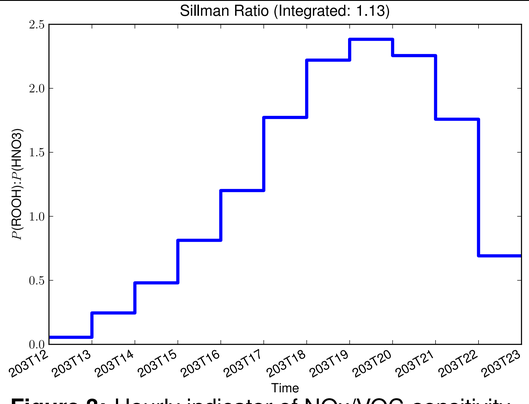


Figure 3: Hourly indicator of NOx/VOC-sensitivity.

RADICAL SOURCE EXAMPLE

Scripted Operation

```
$ python -m perm -c cb05_cmaq test.mrg.nc radical.py
```

Listing 2: radical.py

```
Radical = OH+HO2+MEO2+HCO3+C2O3+CXO3+XO2+TO2+XO2N
subst_net_rxn(HONO, HONO, logical_and = False)
subst_net_rxn(PNA, PNA, logical_and = False)
subst_net_rxn(PAN, PAN, logical_and = False)
subst_net_rxn(PANX, PANX, logical_and = False)
subst_net_rxn(OID, OID, logical_and = False)
subst_net_rxn(ROR, ROR, logical_and = False)
subst_net_rxn(HCO3, HCO3, logical_and = False)

plot_rxns(-Radical, Radical, plot_spc = Radical, \
  title = 'Radical Initiation', path = \
  'radical_init.pdf', chem = None, nlines = 10, \
  axis_settings = dict(xscale = 'log'))
```

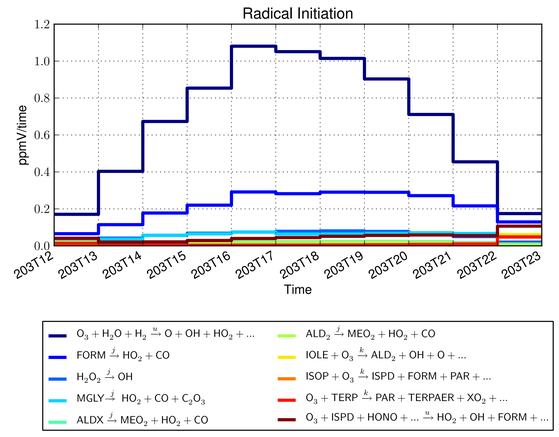


Figure 4: Initiation of radical by reaction.

MORE INFORMATION

To learn more about PERM, go to our website. The website contains detailed installation instructions, extensive tutorials, and a way to provide feedback. For even more detailed information, PERM has extensive documentation. For questions about any command shown here, type help(*command*). For questions about available commands, type help(mech).

Contact:

website: https://dawes.sph.unc.edu/trac/PERM
email: barronh@gmail.com

Acknowledgments

PERM builds on work from many other researchers. In additon to the authors, work by Gail Tonnesen and Jerry Gibson provided much inpsiration and some code.

This research was supported in part by an appointment to the Research Participation Program at the National Exposure Research Laboratory, U.S. Environmental Protection Agency administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and EPA.