Willkommen
Welcome
Bienvenue

Empa
Materials Science and Technology

COSMO & Civil Service

David Ochsner

21.01.2019

Laboratory for Air Pollution/Environmental Technology, Swiss Federal Laboratories for Materials Science and Technology, Empa, Dübendorf, Switzerland

Processing Chain for COSMO-ART

Empa
Materials Science and Technology

- Processing Chain automates running simulations in COSMO
- -> more information in talk from Michael Jähn @ 16:00

```
(amrs) ~/dao_processing_chain/cosmo_processing_chain> python run_chain.py example_cosmoart
_mother example_cosmoart_nested 2015-06-26 0 12 -j meteo icbc emissions obs_nudging photo_
rate int2lm cosmo post_cosmo verify_chain -f
Starting chain for case example_cosmoart_mother, using COSMOART
Process "meteo" for chain "2015062600_0_12"
Process "icbc" for chain "2015062600_0_12"
Process "emissions" for chain "2015062600_0_12"
Process "obs_nudging" for chain "2015062600_0_12"
Process "photo_rate" for chain "2015062600_0_12"
Process "int2lm" for chain "2015062600_0_12"
Submitted batch job 11396563
```

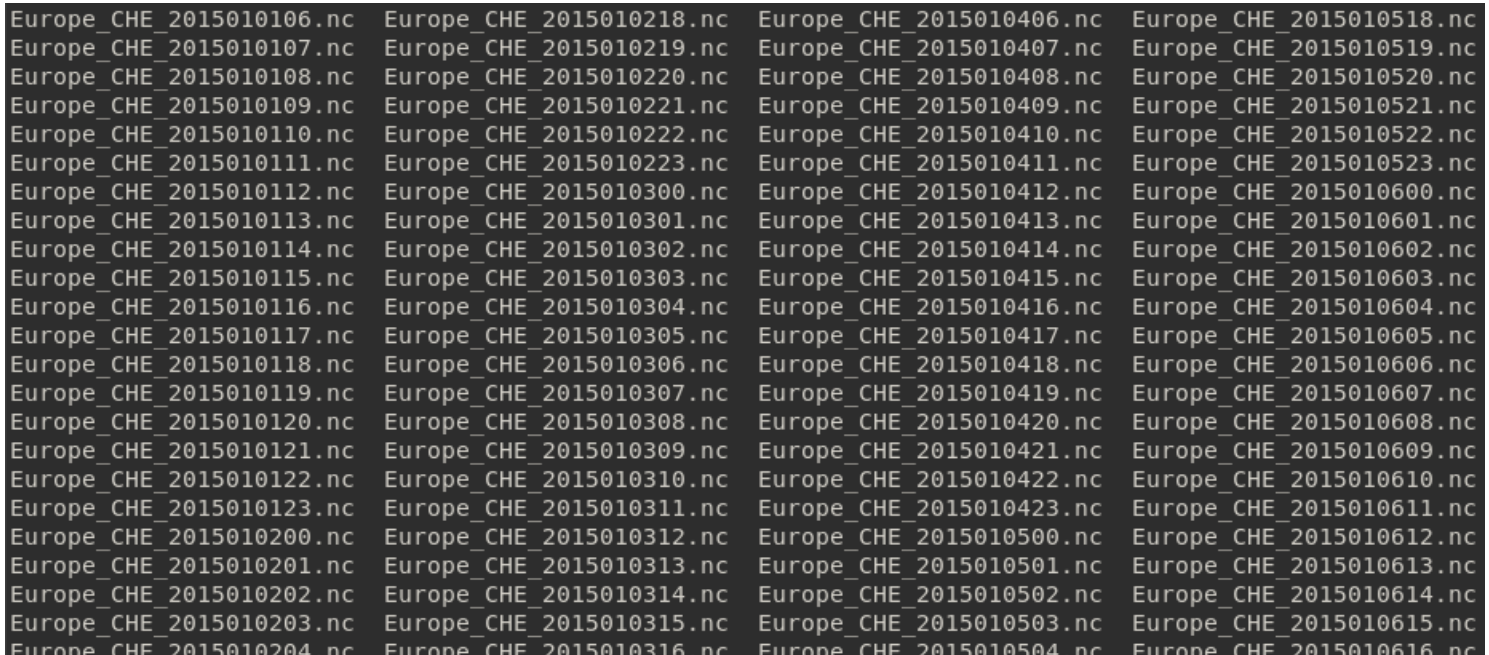
Processing Chain for COSMO-ART

Empa
Materials Science and Technology

- Processing Chain automates running simulations in COSMO(-GHG)
- Extend it to work with COSMO-ART (Qing Mu @ 14:15)
  - Multiple emission datasets
  - Observation nudging
  - Nested domains
  - Documentation
  - Compare results to previous versions

Online Emissions

- Simulating tracers in COSMO-GHG & COSMO-ART:
- Offline Emissions: Read in hourly emissions from preprocessed files



Europe_CHE_2015010106.nc	Europe_CHE_2015010218.nc	Europe_CHE_2015010406.nc	Europe_CHE_2015010518.nc
Europe_CHE_2015010107.nc	Europe_CHE_2015010219.nc	Europe_CHE_2015010407.nc	Europe_CHE_2015010519.nc
Europe_CHE_2015010108.nc	Europe_CHE_2015010220.nc	Europe_CHE_2015010408.nc	Europe_CHE_2015010520.nc
Europe_CHE_2015010109.nc	Europe_CHE_2015010221.nc	Europe_CHE_2015010409.nc	Europe_CHE_2015010521.nc
Europe_CHE_2015010110.nc	Europe_CHE_2015010222.nc	Europe_CHE_2015010410.nc	Europe_CHE_2015010522.nc
Europe_CHE_2015010111.nc	Europe_CHE_2015010223.nc	Europe_CHE_2015010411.nc	Europe_CHE_2015010523.nc
Europe_CHE_2015010112.nc	Europe_CHE_2015010300.nc	Europe_CHE_2015010412.nc	Europe_CHE_2015010600.nc
Europe_CHE_2015010113.nc	Europe_CHE_2015010301.nc	Europe_CHE_2015010413.nc	Europe_CHE_2015010601.nc
Europe_CHE_2015010114.nc	Europe_CHE_2015010302.nc	Europe_CHE_2015010414.nc	Europe_CHE_2015010602.nc
Europe_CHE_2015010115.nc	Europe_CHE_2015010303.nc	Europe_CHE_2015010415.nc	Europe_CHE_2015010603.nc
Europe_CHE_2015010116.nc	Europe_CHE_2015010304.nc	Europe_CHE_2015010416.nc	Europe_CHE_2015010604.nc
Europe_CHE_2015010117.nc	Europe_CHE_2015010305.nc	Europe_CHE_2015010417.nc	Europe_CHE_2015010605.nc
Europe_CHE_2015010118.nc	Europe_CHE_2015010306.nc	Europe_CHE_2015010418.nc	Europe_CHE_2015010606.nc
Europe_CHE_2015010119.nc	Europe_CHE_2015010307.nc	Europe_CHE_2015010419.nc	Europe_CHE_2015010607.nc
Europe_CHE_2015010120.nc	Europe_CHE_2015010308.nc	Europe_CHE_2015010420.nc	Europe_CHE_2015010608.nc
Europe_CHE_2015010121.nc	Europe_CHE_2015010309.nc	Europe_CHE_2015010421.nc	Europe_CHE_2015010609.nc
Europe_CHE_2015010122.nc	Europe_CHE_2015010310.nc	Europe_CHE_2015010422.nc	Europe_CHE_2015010610.nc
Europe_CHE_2015010123.nc	Europe_CHE_2015010311.nc	Europe_CHE_2015010423.nc	Europe_CHE_2015010611.nc
Europe_CHE_2015010200.nc	Europe_CHE_2015010312.nc	Europe_CHE_2015010500.nc	Europe_CHE_2015010612.nc
Europe_CHE_2015010201.nc	Europe_CHE_2015010313.nc	Europe_CHE_2015010501.nc	Europe_CHE_2015010613.nc
Europe_CHE_2015010202.nc	Europe_CHE_2015010314.nc	Europe_CHE_2015010502.nc	Europe_CHE_2015010614.nc
Europe_CHE_2015010203.nc	Europe_CHE_2015010315.nc	Europe_CHE_2015010503.nc	Europe_CHE_2015010615.nc
Europe_CHE_2015010204.nc	Europe_CHE_2015010316.nc	Europe_CHE_2015010504.nc	Europe_CHE_2015010616.nc

Online Emissions

- Simulating tracers in COSMO-GHG & COSMO-ART:
- Online Emissions: Read in yearly emissions & profiles and calculate emissions online

```
dayofweek.nc
Europe_CHE_2015.nc
hourofday.nc
monthofyear.nc
vertical_profiles.nc
```

Online Emissions for COSMO-ART

Empa
Materials Science and Technology

- Online Emissions: Already implemented in COSMO-GHG
- Port to COSMO-ART
- Add support for speciation: Emission categories can partially contribute to a tracer.

```
OFFENDING LINES IN /users/ochsnerd/cosmo-pompa/cosmo/build.log:
ftn-855 crayftn: ERROR SRC_BIOGEOCHEM, File = ../../../users/ochsnerd/cosmo-pompa/cosmo/s
rc/src_biogeochem.f90, Line = 4, Column = 8
  The compiler has detected errors in module "SRC_BIOGEOCHEM".  No module information fil
e will be created for this module.
ftn-297 crayftn: ERROR INPUT_TRACER, File = ../../../users/ochsnerd/cosmo-pompa/cosmo/src
/src_biogeochem.f90, Line = 788, Column = 24
ftn-508 crayftn: ERROR INPUT_TRACER, File = ../../../users/ochsnerd/cosmo-pompa/cosmo/src
/src_biogeochem.f90, Line = 866, Column = 5
ftn-297 crayftn: ERROR INPUT_TRACER, File = ../../../users/ochsnerd/cosmo-pompa/cosmo/src
/src_biogeochem.f90, Line = 902, Column = 5
ftn-724 crayftn: ERROR INPUT_TRACER, File = ../../../users/ochsnerd/cosmo-pompa/cosmo/src
/src_biogeochem.f90, Line = 902, Column = 15
Cray Fortran : 5 errors, 0 warnings, 0 other messages, 0 ansi
/users/ochsnerd/cosmo-pompa/cosmo/Makefile:282: recipe for target 'src_biogeochem.o' fail
ed
```

Python & netCDF

- Transforming netCDF-files in python using netCDF4 can be cumbersome

```python
from netCDF4 import Dataset

# Add two variables and write to new file
with Dataset('file1.nc') as src, Dataset('file2.nc', 'a') as dst:
    dst.createVariable('varname12',
                       src['varname1'].dtype,
                       src['varname1'].dimensions)
    dst['varname12'][:] = src['varname1'][:] + src['varname2'][:]
    for ncattr in src['varname1'].ncattrs():
        dst['varname12'].setncattr(name=ncattr,
                                   value=src['varname1'].getncattr(ncattr))


# Extract a subset of a given dataset into a new one
with Dataset('file1.nc') as src, Dataset('file2.nc', 'w') as dst:
    # Copy all dimensions, but some are smaller
    for dim in src.dimensions.keys():
        if dim == 'rlat':
            dst.createDimension(dim, 100)
        elif:
            dst.createDimension(dim, 100)
        else:
            dst.createDimension(dim, src.dimensions[dim].size)

    # Copy variables. Some are smaller, some not
    for var in src.variables.keys():
        # only works if variable has ALL or NONE of the reduced dimensions
        if any(dim in ['rlat', 'rlon'] for dim in src[var]):
            dst.createVariable(var,
                               src[var].dtype,
                               src[var].dimensions)
            dst[var][:] = src[var][:, 150:250, 100:200]
            for ncattr in src['varname'].ncattrs():
                dst['varname'].setncattr(name=ncattr,
                                         value=src['varname'].getncattr(ncattr))
        else:
            dst.createVariable(var,
                               src[var].dtype,
                               src[var].dimensions)
            dst[var][:] = src[var][:]
            for ncattr in src['varname'].ncattrs():
                dst['varname'].setncattr(name=ncattr,
                                         value=src['varname'].getncattr(ncattr))
```

Python & netCDF

Empa
Materials Science and Technology

```python
from netCDF4 import Dataset

from nc_operations import copy_variable, extract_subdomain

# Add two variables and write to new file
with Dataset('file1.nc') as src, Dataset('file2.nc', 'a') as dst:
    copy_variable(src, dst,
                  src_names=['varname1', 'varname2'],
                  dst_name='varname12')


# Extract a subset of a given dataset into a new one
with Dataset('file1.nc') as src, Dataset('file2.nc', 'w') as dst:
    extract_subdomain(src, dst,
                      {'rlat': slice(100, 200, 1),
                       'rlon': slice(150, 250, 1)})
```

```python
from netCDF4 import Dataset

# Add two variables and write to new file
with Dataset('file1.nc') as src, Dataset('file2.nc', 'a') as dst:
    dst.createVariable('varname12',
                       src['varname1'].dtype,
                       src['varname1'].dimensions)
    dst['varname12'][:] = src['varname1'][:] + src['varname2'][:]
    for ncattr in src['varname1'].ncattrs():
        dst['varname12'].setncattr(name=ncattr,
                                   value=src['varname1'].getncattr(ncattr))


# Extract a subset of a given dataset into a new one
with Dataset('file1.nc') as src, Dataset('file2.nc', 'w') as dst:
    # Copy all dimensions, but some are smaller
    for dim in src.dimensions.keys():
        if dim == 'rlat':
            dst.createDimension(dim, 100)
        elif:
            dst.createDimension(dim, 100)
        else:
            dst.createDimension(dim, src.dimensions[dim].size)

    # Copy variables. Some are smaller, some not
    for var in src.variables.keys():
        # only works if variable has ALL or NONE of the reduced dimensions
        if any(dim in ['rlat', 'rlon'] for dim in src[var]):
            dst.createVariable(var,
                               src[var].dtype,
                               src[var].dimensions)
            dst[var][:] = src[var][:, 150:250, 100:200]
            for ncattr in src['varname'].ncattrs():
                dst['varname'].setncattr(name=ncattr,
                                         value=src['varname'].getncattr(ncattr))
        else:
            dst.createVariable(var,
                               src[var].dtype,
                               src[var].dimensions)
            dst[var][:] = src[var][:]
            for ncattr in src['varname'].ncattrs():
                dst['varname'].setncattr(name=ncattr,
                                         value=src['varname'].getncattr(ncattr))
```

- Library to simplify common tasks?