Neighborhood Emission Mapping Operation (NEMO)

A high-resolution dataset of anthropogenic emissions

HOW TO READ THE DATA FILES

Variables

The NetCDF provides the model species for CB6 mechanism in the hourly and monthly data files with unit of moles/s. The species information can be found in the species table: https://github.com/USEPA/CMAQ/blob/5.3.3/CCTM/src/MECHS/mechanism_information/cb6r3_ae7_aq

The annual emissions in NetCDF, shapefiles and csv (point source) include VOCs (VOCANN), NO_x (NOXANN), $PM_{2.5}$ (PM25ANN), SO_2 (SO2ANN), CO (COANN), and NH_3 (NH3ANN) with unit of ton/km^2/yr.

VOCANN contains the reactive species in CB6: ACET, ACROLEIN, ALD2, ALDX, BENZ, BUTADIENE13, CH4, ETH, ETHA, ETHY, ETOH, FORM, IOLE, ISOP, KET, MEOH, NAPH, OLE, PAR, PRPA, FACD, TERP, TOL, XYLMN, AACD, APIN

NOXANN contains: NO, NO2, HONO

PM25ANN contains: PEC, PNO3, POC, PSO4, PAL, PCA, PCL, PFE, PH2O, PK, PMG, PMN, PMOTHR, PNA, PNH4, PNCOM, PSI, PTI, PMC

*The unit conversion from mole to ton is done by multiplying the species emissions in moles with their molecular weights (shown in the species table).

How to Read Shapefiles (.shp) Format

Files you need: {filename}.cpg, {filename}.dbf, {filename}.prj, {filename}.shp, {filename}.shx

{filename} is the name of the shapefile

Each .shp file uses the same .cpg and .prj files, which are not included in us01emis_annual_shp.zip of https://doi.org/10.6084/m9.figshare.c.6141735. Users can create two new text files, rename the files to {filename}.cpg and {filename}.prj, and add the contents shown below, respectively:

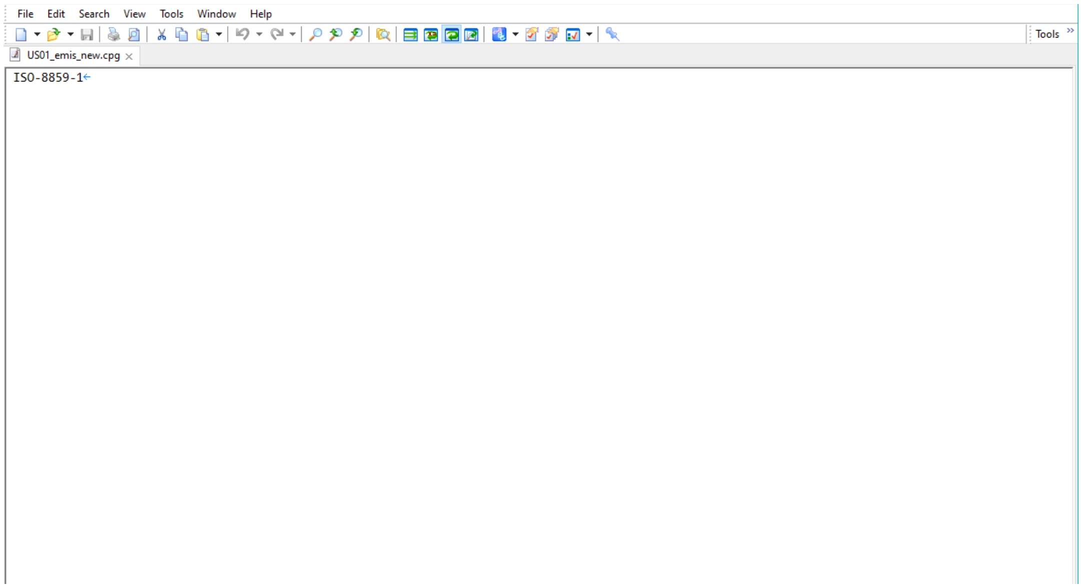
{filename}.cpg:

ISO-8859-1

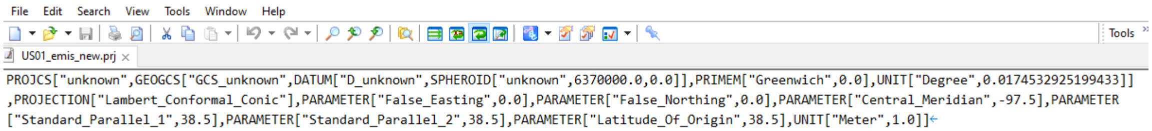
{filename}.prj:

PROJCS["unknown",GEOGCS["GCS_unknown",DATUM["D_unknown",SPHEROID["unknown",6370000.0,0.0]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Lambert_Conformal_Conic"],PARAMETER["False_Easting",0.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-97.5],PARAMETER["Standard_Parallel_1",38.5],PARAMETER["Standard_Parallel_2",38.5],PARAMETER["Latitude_Of_Origin",38.5],UNIT["Meter",1.0]]

They will show like this:



```
ISO-8859-1
```

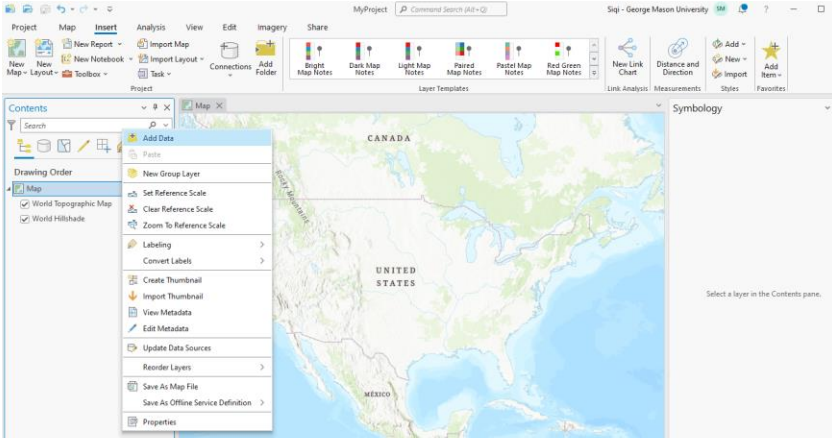


```
PROJCS["unknown",GEOGCS["GCS_unknown",DATUM["D_unknown",SPHEROID["unknown",6370000.0,0.0]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]]
,PROJECTION["Lambert_Conformal_Conic"],PARAMETER["False_Easting",0.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-97.5],PARAMETER
["Standard_Parallel_1",38.5],PARAMETER["Standard_Parallel_2",38.5],PARAMETER["Latitude_Of_Origin",38.5],UNIT["Meter",1.0]]
```

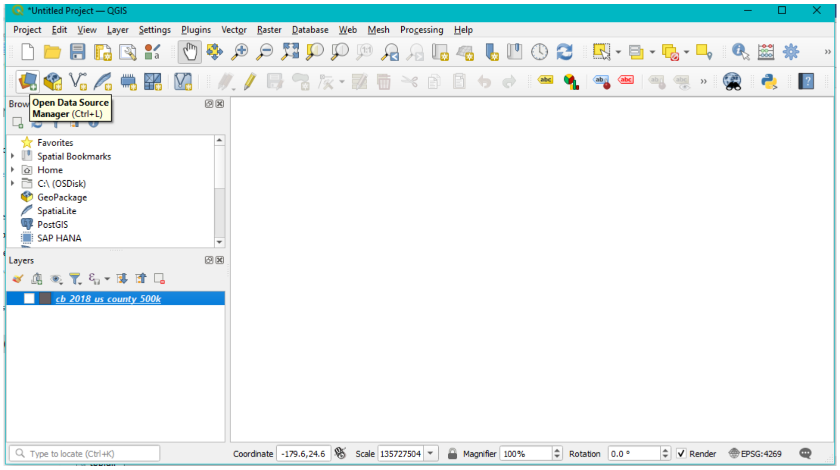
1. *GIS: ArcGIS, QGIS, etc.*

Drag {filename}.shp to the software (e.g., ArcMap, QGIS) to open. For ArcGIS, the “maximum sample value” needs to be set to a larger number. For the CONUS scale plot, recommend using QGIS, since **ArcGIS may show errors on the boundaries due to the large sampling grids**.

ArcMap:



QGIS:



2. *Programming: python.*

```
import shapefile
input = shapefile.Reader(“shpfilename”)
fields = input.field[1:] # get field
fields_name = [field[0] for field in fields] # get field name
for rec in input.iterShapeRecords(): # get records by line
  col = rec.record[0]
  row = rec.record[1]
  …
```

3. *Programming: NCL.*

Instructions are provided by NCL at https://www.ncl.ucar.edu/Applications/shapefiles.shtml

How to Read NetCDF (.ncf) Format

The files are written in NetCDF format by IOAPI (https://www.cmascenter.org/ioapi/) with surfix .ncf. Most programming languages can read them.

The projection information is shown in global attributes:

```
        :GDTYP = 2 ;            ! means LCC projection
        :P_ALP = 38.5 ;         !  true latitude 1
        :P_BET = 38.5 ;         !   true latitude 2
        :P_GAM = -97.5 ;        !   standard longitude
        :XCENT = -97.5 ;        !  central point longitude
        :YCENT = 38.5 ;         ! central point latitude
        :XORIG = -2701000.25 ;    ! x-axis start point (lower-left) in meters
        :YORIG = -1580581.38 ;     ! y-axis start point (lower-left) in meters
        :XCELL = 1000. ;                 ! x-axis resolution in meters
        :YCELL = 1000. ;                  ! y-axis resolution in meters
```

1. *Read file (fortran.ioapi):*

```
!...........   INCLUDES:
   INCLUDE 'PARMS3.EXT'
   INCLUDE 'IODECL3.EXT'
   INCLUDE 'FDESC3.EXT'
!............ Define variable
   INTEGER   NSTEPS, NVARS, NLAYS, NCOLS, NROWS, TSTEP, SDATE, STIME
   INTEGER   EDATE,  ETIME
   INTEGER   T_LOOP, C_LOOP, R_LOOP, V_LOOP, L_LOOP
   CHARACTER*256 INFILE
   CHARACTER*16  PNAME
   REAL, ALLOCATABLE,SAVE :: VAR_INV (:,:,:,:)   ! variables in file
!...   open the emission file
PNAME = "READ_EMIS" ! name the program
   INFILE = "INFILE" ! input file from environment variable: export INFILE=
   IF ( .NOT. OPEN3( INFILE, FSREAD3, PNAME ) ) THEN
     CALL M3EXIT( 'INFILE1', 0, 0, 'Can not open infile1', 2 )
   END IF
!...   Extract global variables and descriptions;
   IF ( .NOT. DESC3( INFILE ) ) THEN
     PRINT *, "  **  Can not get file1 description"
   END IF
   NSTEPS  = MXREC3D
   NVARS   = NVARS3D
   NLAYS   = NLAYS3D
   NCOLS   = NCOLS3D
   NROWS   = NROWS3D
   TSTEP   = TSTEP3D
   EDATE   = SDATE3D
   ETIME   = STIME3D
!...   Read variables by time steps;
   ALLOCATE ( VAR_INV (NCOLS, NROWS, NLAYS, NVARS))
   DO T_LOOP = 1, NSTEPS
     IF ( .NOT. READ3(INFILE, ALLVAR3, ALLAYS3, EDATE,
  &    ETIME, VAR_INV))THEN
        PRINT *, "Reading is not done for 1: ", EDATE
     END IF
   END DO
```

***To get the lon/lat, use the lon/lat to lcc conversion function in IOAPI:**

```
      IF ( .NOT. OPEN3( INFILE, FSREAD3, PNAME ) ) THEN
        CALL M3EXIT( 'INFILE1', 0, 0, 'Can not open infile1', 2 )
      END IF
!...   Extract global variables and descriptions;
      IF ( .NOT. DESC3( INFILE ) ) THEN
```

```
        PRINT *, "  **  Can not get file1 description"
      END IF
      IF( .NOT. SETLAM( SNGL( P_ALP3D),     !  define projection
     &                  SNGL( P_BET3D),
     &                  SNGL( P_GAM3D),
     &                  SNGL( XCENT3D),
     &                  SNGL( YCENT3D)  ) ) THEN
        MSG = 'Lambert projection setup error'
        WRITE(*,*) MSG
        CALL M3EXIT( PNAME, 0, 0, MSG, 2 )
      ENDIF
        IF( .NOT. LAM2LL( X, Y, LON, LAT) ) THEN
          MSG = 'Lambert to Lat/Lon conversion error'
          CALL M3EXIT( PNAME, 0, 0, MSG, 2 )
        END IF
```

** IOAPI tutorial: https://www.cmascenter.org/ioapi/documentation/all_versions/html/TUTORIAL.html*

2. ***Read file (python):***

```
from netCDF4 import Dataset
import netCDF4 as nc
import os
env_dist  = os.environ
emisfile = env_dist.get(‘emisfile’)
emis_fid = Dataset(emisfile, 'r')
var_in = emis_fid.variables['VARNAME']  # VARNAME is the name of variable
```

***To get the lon/lat, it needs to transfer the coordination of lcc projection to lon/lat:**

```
from pyproj import Proj, Transformer
xorg = -2701000.25
```

```
yorg = -1580581.38

m_cols = 5397

m_rows = 3177

p_alp = 38.5

p_bet = 38.5

p_gam = -97.5

xcent = -97.5

ycent = 38.5

xcell = 1000.0

ycell = 1000.0

projstr = +proj=lcc +lat_1=38.5 +lat_2=38.5 +lat_0=38.5 +lon_0=-97.5 +x_0=0 +y_0=0 +a=6370000 +b=6370000 +units=m +no_defs"

transformer = Transformer.from_crs(projstr,"epsg:4326")

pdom = Proj(projstr)

xmin = xorg

ymin = yorg

xmax = xorg + (m_cols-1) * xcell

ymax = yorg + (m_rows-1) * ycell

lat = np.zeros((ny, nx), dtype=float)

lon = np.zeros((ny,nx), dtype=float)

nn = 0

for col in range(nx) :

  for row in range(ny) :

    xx = xorg + col * xcell+ xcell/2

    yy = yorg + row * ycell+ ycell/2

    lon[row,col],lat[row,col] = pdom(xx,yy,inverse=True)

    print(xx,yy,lon[row,col],lat[row,col])
```

3. *Read file (NCL):*

Instructions are provided by NCL at https://www.ncl.ucar.edu/Applications/netcdf4.shtml

4. *Visualization tools (VERDI, Panoply etc.)*

VERDI (https://www.cmascenter.org/verdi): reduce the time or lat/lon dimension sizes to avoid the memory exceedance.

Panoply (https://www.giss.nasa.gov/tools/panoply): Panoply can only recognize the suffix .nc for the NetCDF file, change the .ncf to .nc to use.