Running the Surrogate Tools DB

Prerequisites:

PostgreSQL Database

Knowledge of GIS/SQL/PostgreSQL SQl is very useful.

GDAL: https://gdal.org/

Overview:

SurrogateToolsDB uses user specified shapefiles and fields to spatially allocate emission inventory data to a user defined grid.

The general workflow using SurrogateToolsDB is:

1. Define modeling grid projection in the PostgreSQL database.
2. Generate a modeling grid and load it into the database.
3. Load data shapefiles and reproject
4. Create ancillary files.
5. Create surrogates.
6. Check logs.
7. Gapfill surrogates.
8. Check logs and generate quality assurance reports.

How To Create Surrogates using SurrogateToolsDB:

1. Define a modeling grid projection

In the PostgreSQL database, define the projection of your modeling grid. This is done by adding a record to the "spatial_ref_sys" table. An example of adding a commonly used projection to the database, used for grids such as 12US1 and 36US3, can be found here https://github.com/CEMPD/SurrogateToolsDB/blob/v1.3/util/create_900921.sql. Projections can be reused to develop multiple surrogates. Note, most grids use the WRF-ARW sphere, assuming the earth has a constant radius of 6370km.

2. Generate and load a modeling grid

Modeling grids should be defined using IO/API conventions for modeling grids: https://www.cmascenter.org/ioapi/documentation/all_versions/html/GRIDDESC.html. The provided GRIDDESC.txt shows what the GRIDDESC should look like for the 12US1 modeling domain. Note, that the 12US1 domain for generating surrogates is intentionally larger than the actual 12US1 SMOKE domain.

Once the grid projection has been defined, the grid needs to be defined and added to the PostgreSQL database as a table. Surrogate grids should be larger than the modeling grid to ensure emissions are not improperly allocated. Inventories are usually county specific. Therefore, all counties within the bounds of the modeling domain need to be completely covered by the surrogate domain. Failing to do will result in possibly large errors due to SMOKE automatically renormalizing county surrogate fractions to add to one. A SurrogateToolsDB utility script already exists to help users create a rectilinear grid in a PostgreSQL database:

https://github.com/CEMPD/SurrogateToolsDB/blob/v1.3/util/generate_modeling_grid.sh

This script is setup to create a rectangular grid table consistent with the 12US1 domain. Non-rectilinear grids can be developed either by creating a shapefile or a database table to represent the grid. In the case that a desired grid is in a shapefile format and the utility script is not being used, the shapefile can be loaded into the database using ogr2ogr as follows, where <GRID_SHAPEFILE> is a shapefile that contains your grid geometry and <GRIDNAME> is the desired name of the grid table in the database:

```
ogr2ogr -f "PostgreSQL" "PG:dbname=<YOUR_DATABASE_NAME>
user=<YOUR_DATABASE_USER> host=<YOUR_DATABASE_HOST>"
<GRID_SHAPEFILE> -lco PRECISION=NO -nlt PROMOTE_TO_MULTI -nln
public.<GRIDNAME> -overwrite
```

Note that surrogate grids in the respective PostgreSQL database tables need to use the same column naming conventions as in the utility script (generate_modeling_grid.sh).

It is highly recommended to visualize your modeling domain to ensure the desired counties are fully covered by the domain. PostgreSQL database tables can be exported as shapefiles through command line utilities, such as pgsql2shp.

Once you are satisfied with the domain, add projection and domain information to the GRIDDESC.txt file. GRIDDESC.txt files can support multiple grid definitions in a single file.

3. Load data shapefiles and reproject

   All data shapefiles need to be loaded into the PostgreSQL database and reprojected to the modeling domain. This is usually done by first using the command line utility ogr2ogr to load the shapefile into the database. Once the data are loaded, the database table that contains the shapefile data must be reprojected to the projection of the desired modeling domain defined in the "spatial_ref_sys" table. Optionally, other fields used to create surrogates can, also, be added during this step. The commmand "psql" with SQL commands can be used to reproject and manipulate data tables.

   For examples, see the provided scripts "load_shapefile.csh" and "load_shapefile_reproject.csh".

4. Create ancillary files

   The ancillary files used to run the SurrogateToolsDB are defined at https://github.com/CEMPD/SurrogateToolsDB/tree/v1.3/docs under "Input Files". There are 5 ancillary files that are needed.

   - "surrogate_codes.csv" lists all surrogate code and names.
   - "shapefile_catalog.csv" describes all shapefiles used and their respective projections.
   - "surrogate_specification.csv" file tells SurrogateToolsDB how to generate each surrogate and what data to use.

   Within the surrogate specification file, the "DATA SHAPEFILE" is usually a shapefile that contains county geometries. Data Shapefiles for U.S. modeling are normally based on Census TIGER county shapefiles available at https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html.

   The "WEIGHT SHAPEFILE" is the shapefile from which data are apportioned to counties and grid cells to generate the surrogate ("NOFILL").

   The SECONDARY/TERTIARY/QUARTERNARY SURROGATE fields denote the surrogates to gapfill the surrogate ("FILL"), where data for a particular county (or county equivalent) are not available in the weight shapefile.

   The "surrogate_generation.csv" instructs SurrogateToolsDB to create and perform quality assurance on specific surrogates.

   The "PG SCRIPT" field declares the PostgreSQL script that will be used to create a surrogate. Different scripts are used for point, line, and polygon data. The PG scripts are available at https://github.com/CEMPD/SurrogateToolsDB/tree/master/pgscripts. Note the naming conventions "withFF"/"noFF" and "withWA"/"noWA". "FF" stands for "FILTER FUNCTION" and "WA" stands for "WEIGHT ATTRIBUTE". These names correspond to the fields in "surrogate_specification.csv" and if these fields are being used to generate the surrogate.

The "control_variables.csv" file sets the grid, projection, and other settings SurrogateToolsDB should use. Gapfilling is also controlled in this script. Set "COMPUTE SURROGATES FROM SHAPEFILE" and "COMPUTE SURROGATES" to "NO" to generate surrogates prior to gapfilling. Then, set "MERGE SURROGATES" to create surrogates merged from existing surrogates and set "GAPFILL SURROGATES" to "YES" to gapfill surrogates. Details of these ancillary files can be found on the SurrogateToolsDB GitHub page.

Ancillary files for EPA platforms can be found in Excel spreadsheets. For the 2020 platform, these files are included in "2020_Spatial Surrogates Development with CANMEX_101023.xlsx" at https://gaftp.epa.gov/Air/emismod/2020/spatial_surrogates/.

5. Generate surrogates without gapfilling.

   First, edit control_variables.csv so that "COMPUTE SURROGATES FROM SHAPEFILE" and "COMPUTE SURROGATES" set to "YES"; "MERGE SURROGATES" and "GAPFILL SURROGATES" set to "NO".

   Next, edit "srgcreate.csh" so that "java -classpath <PATH>/SurrogateTools-2.2.jar gov.epa.surrogate.ppg.Main control_variables.csv" is uncommented. Any other Java command should be commented out.

   Finally, run srgcreate.csh without gapfilling. This step may take a few days to run depending on the input data and domain size.

6. Check logs

   Check the log in ./LOGS. Check to make sure each surrogate that was supposed to be created did not fail. If one surrogate failed and others succeeded, the "surrogate_generation.csv" can be edited and "srgcreate.csh" can be rerun to only create specific surrogates. Created surrogates should be in ./outputs/<GRIDNAME> with "NOFILL" in the name.

7. Gapfill surrogates.

   Edit "control_variables.csv" so that "COMPUTE SURROGATES FROM SHAPEFILE" and "COMPUTE SURROGATES" set to "NO"; "MERGE SURROGATES" and "GAPFILL SURROGATES" set to "YES". Edit "srgcreate.csh" so that "java -classpath <PATH>/SurrogateTools-2.2.jar gov.epa.surrogate.SurrogateTool control_variables.csv" is uncommented. Any other Java command should be commented out. Then run "srgcreate.csh" this time for gapfilling. This step should take a few minutes to run or less.

8. Check logs and generate quality assurance reports.

   Check the newly generated logs in ./LOGS to see if gapfilling was sucessful. Gapfilled surrogates should be located in ./outputs/<GRIDNAME> with "FILL" in the name.

   Next, generate quality assurance reports by editing "qa_srg.csh" to point to the created surrogate description "srgdesc*" file in ./outputs, and run "qa_srg.csh". The "qa_surg.csh" script will generate several CSV reports. Details about each of these files can be found on the SurrogateToolsDB Github site https://github.com/CEMPD/SurrogateToolsDB/tree/master.

Data files and scripts for the 2021 platform:

Surrogates for the 2021 platform use the same shapefiles as the 2020 platform along with shapefiles with 2021 oil and gas data. The Shapefiles and other data files can be found at

https://gaftp.epa.gov/Air/emismod/2020/spatial_surrogates/ for 2020 platform and https://gaftp.epa.gov/air/emismod/2021/spatial_surrogates/ for 2021 oil and gas and the updated Mexico population surrogate.

The 2020 platform ancillary files needed to run the Surrogate Tools can be found in the 2020 platform ftp link in "2020_Spatial Surrogates Development with CANMEX_101023.xlsx". These same ancillary files are, also, found in ./platform_2020 in this package. Ancillary files for oil and gas surrogates are not currently posted on the FTP site. These files are located in ./oilgas_2021. These are the ancillary files mentioned in step 4. In particular, the "control_variables.csv" file will need to be carefully edited to point to user system files. The provided GRIDDESC.txt is an example GRIDDESC used for generating surrogates for the 12US1 domain.

For step 3, Several scripts useful for loading shapefiles are present in ./platform_2020/shp and ./oilgas_2021/shp. Users will need to edit the "load_shapefile_reproject" scripts to point to shapefiles on their own system and for their own grid projections. The shapefiles listed in these scripts may not be a complete set to run all surrogates for 2021.

In ./platform_2020 and ./oilgas_2021 the upper level script to create and merge surrogates, "srgcreate.csh", and to generate quality assurance reports, "qa_srg.csh", are provided. Please read the comments in these files and setup necessary variables relevant to your own environment before attempting to run steps 5 and 7.

Tips and Tricks:

- In PostgreSQL "\d <TABLENAME>" will yield useful information regarding the columns and column datatypes of a specified table. This command is your friend to confirm whether your data looks like it has been loaded properly and set to the right projection.

- Tables can be exported as shapefiles from the PostgreSQL database using pgsql2shp. This is very useful for visualizing reprojected data and the grid in the PostgreSQL database.

- Authentication issues with the PostgreSQL database may happen while attempting to run the Surrogate Tools DB. Refer to PostgreSQL documentation to see ways to navigate this. Authentication variables like PGPASSWORD may be set in the ./pgscripts/*.csh directly or pg_setup.csh.

- It is up to the user to understand if their data is being properly projected and handled. Shapefiles and other geospatial data can be plotted using ESRI ArcGIS, QGIS, or other open source tools, like Python using Matplotlib and Geopandas. If creating surrogates based on EPA platform data, it can be a very useful quality assurance step to compare the spatial distribution of the fractions of generated surrogates with available 12km and 36km surrogates.

- After an initial run, surrogates for finer or coarser domain or different grid can be easily created so long as the projection of the grid is exactly the same. If the projection of a new modeling grid is exactly the same as a previous run, set the GRIDDESC file and edit the control_variables.csv file to use your new domain. Then, run steps 5 through 6. No need to reload shapefiles.