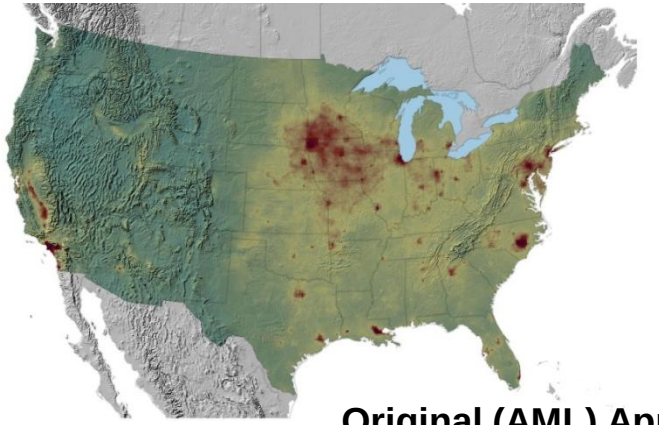
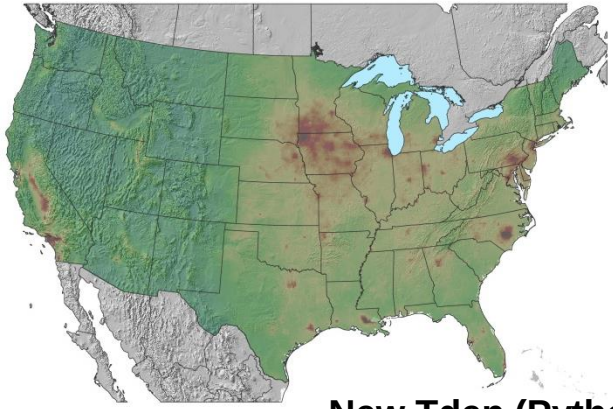
STi | Sonoma Technology

TDep Measurement Model Fusion (MMF) Method to Fuse Modeled and Measured Air Quality Data to Estimate Total Deposition with Python Geoprocessing



Original (AML) Application



New Tdep (Python) Application

By Shih Ying "Changsy" Chang, Nathan Pavlovic, Greg Beachley, Melissa Puchalski, and Christopher Rogers

For the 19th Annual CMAS Conference

October, 2020

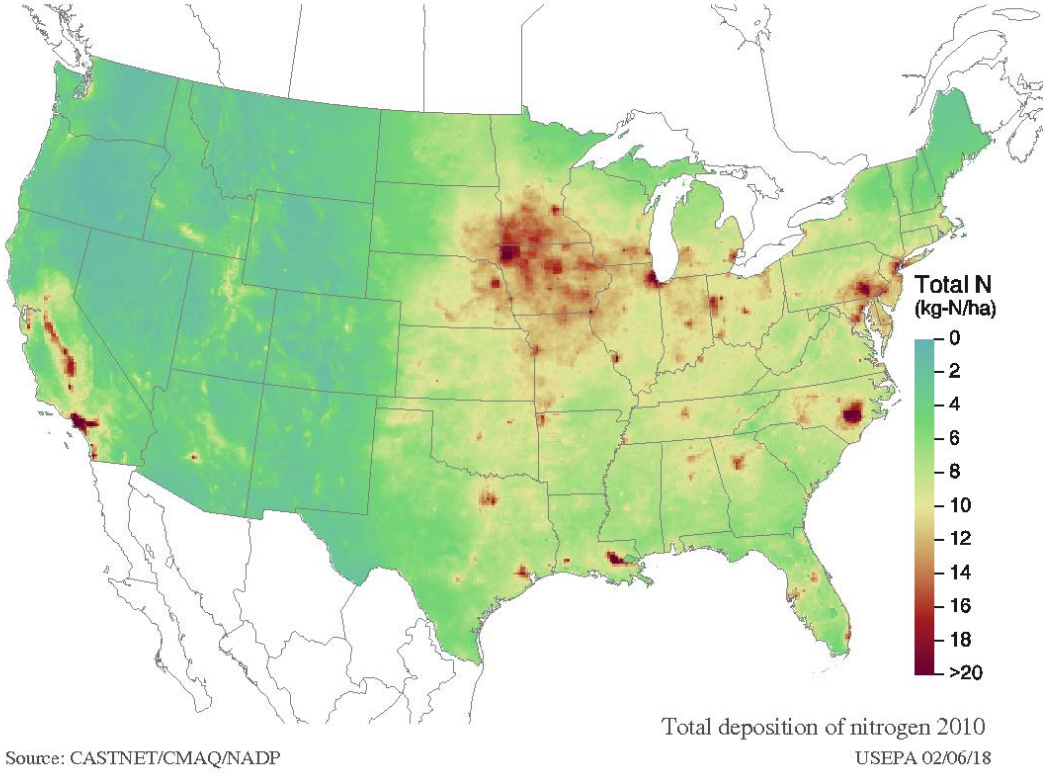
STI-7419

Outline

- Background
- TDep Code philosophy
- Modules in the Python application
- Results
- Summary and future improvements

Background

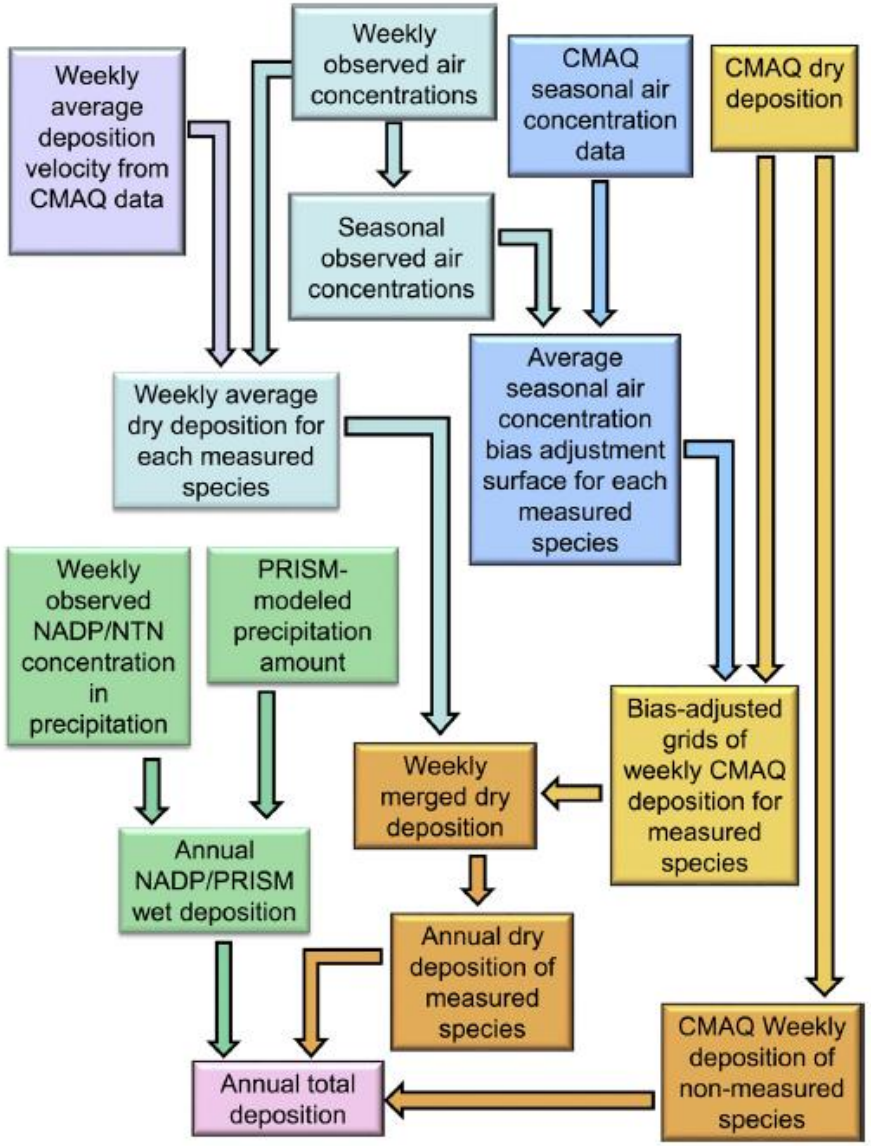
- EPA's **Total Deposition (TDep)** program applies the **Measurement Model Fusion (MMF)** technique to estimate dry and wet deposition for sulfur and nitrogen
  - http://nadp.slh.wisc.edu/committees/tdep/tdepmaps/
- The MMF technique involves complex geoprocessing steps to fuse multiple measured and model datasets



Total nitrogen deposition with the AML script

TDep MMF Technique Overview

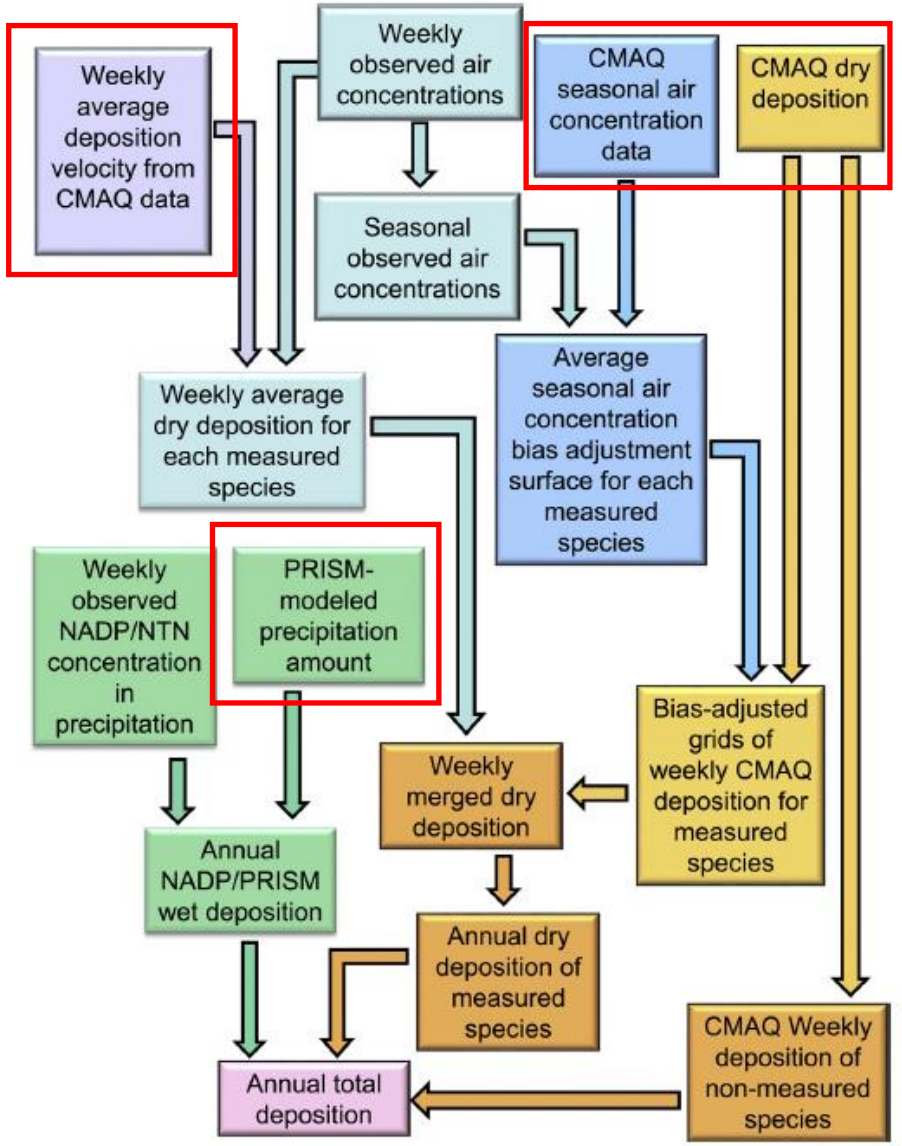
- Modeled data
- Measured data



*Schwede and Lear, *Atmospheric Environment*, 2014

TDep MMF Technique Overview: Modeled Data

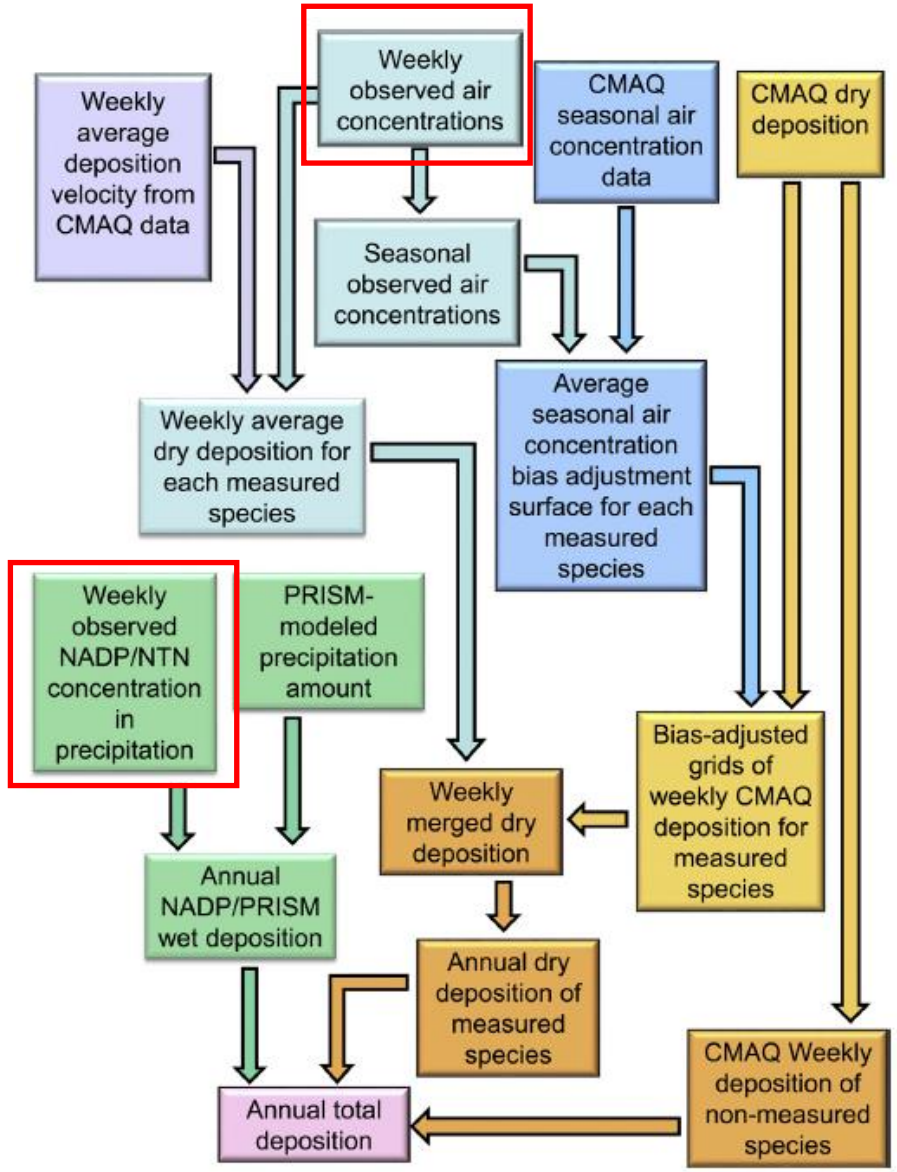
- The Community Multiscale Air Quality System (**CMAQ**)
  - **Dry deposition:** bias adjusted, then fused with the measured dry deposition
  - **Deposition velocity:** combined with measured ambient concentrations to calculate the measured dry deposition
  - **Ambient concentration:** used for adjusting the bias in the modeled dry deposition
- Parameter-elevation Relationships on Independent Slopes Model (**PRISM**)
  - **Precipitation:** fused with the measured precipitation



*Schwede and Lear, *Atmospheric Environment*, 2014

TDep MMF Technique Overview: Measured Data

- Clean Air Status and Trends Network (**CASTNET**)
  - **Ambient concentration:** interpolated, then combined with CMAQ deposition velocity to calculate the measured dry deposition
- National Atmospheric Deposition Program (**NADP**)
  - **Precipitation:** interpolated, then fused with PRISM modeled precipitation
  - **Precipitation chemistry:** interpolated, then combined with fused precipitation field to calculate wet deposition



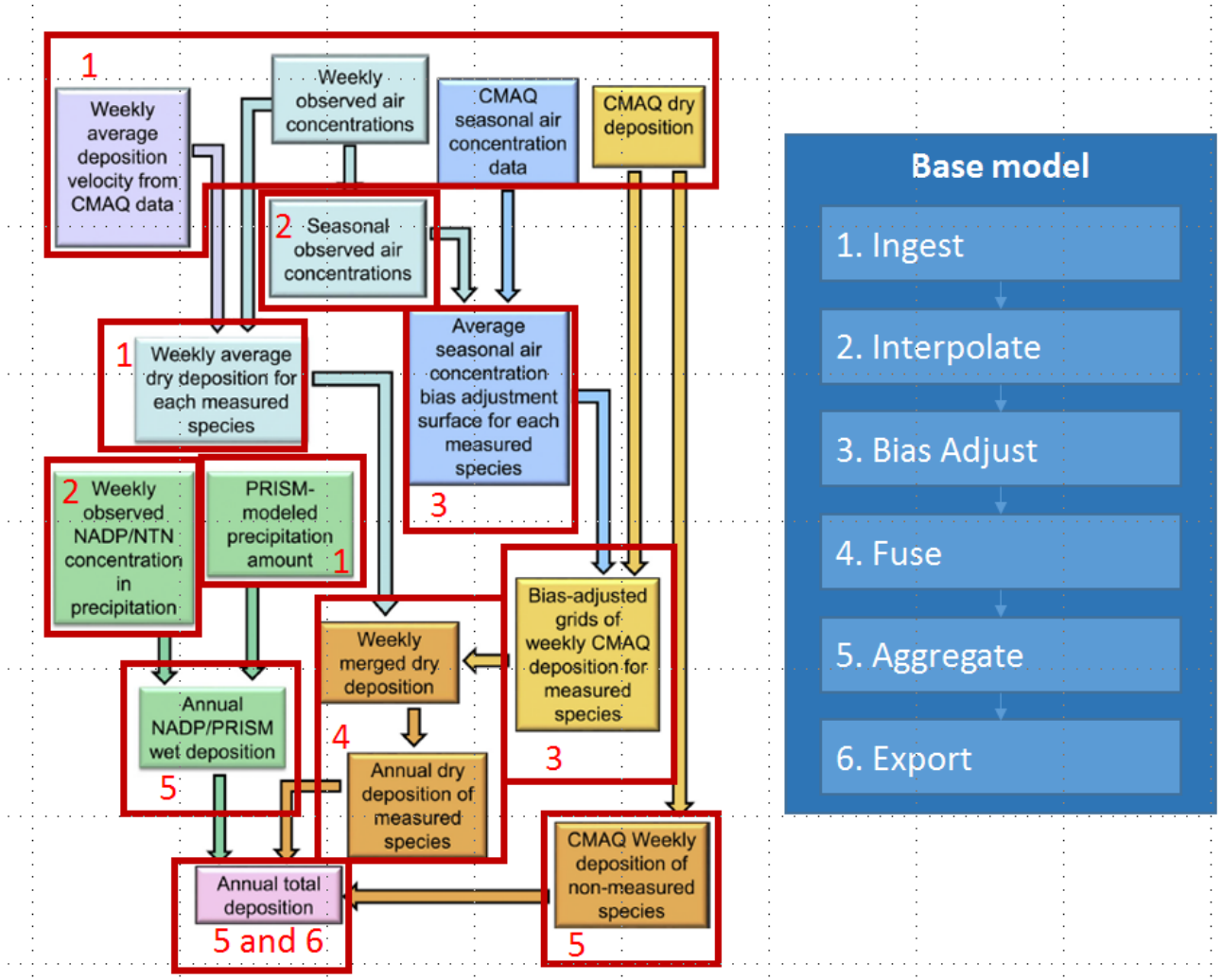
*Schwede and Lear, *Atmospheric Environment*, 2014

Purpose

- The original TDep MMF procedure was implemented with ESRI's ARC Macro Language (AML)
  - Not updated by ESRI anymore
  - Only supports the older ArcGIS platform (version 9.x)
  - Does not facilitate the modern programming practice
  - ESRI replaced AML with **arcpy**, a python library
- Due to the complex geoprocessing steps and multiple datasets, an organized and iterable implementation is required to support ongoing data production and future process enhancements

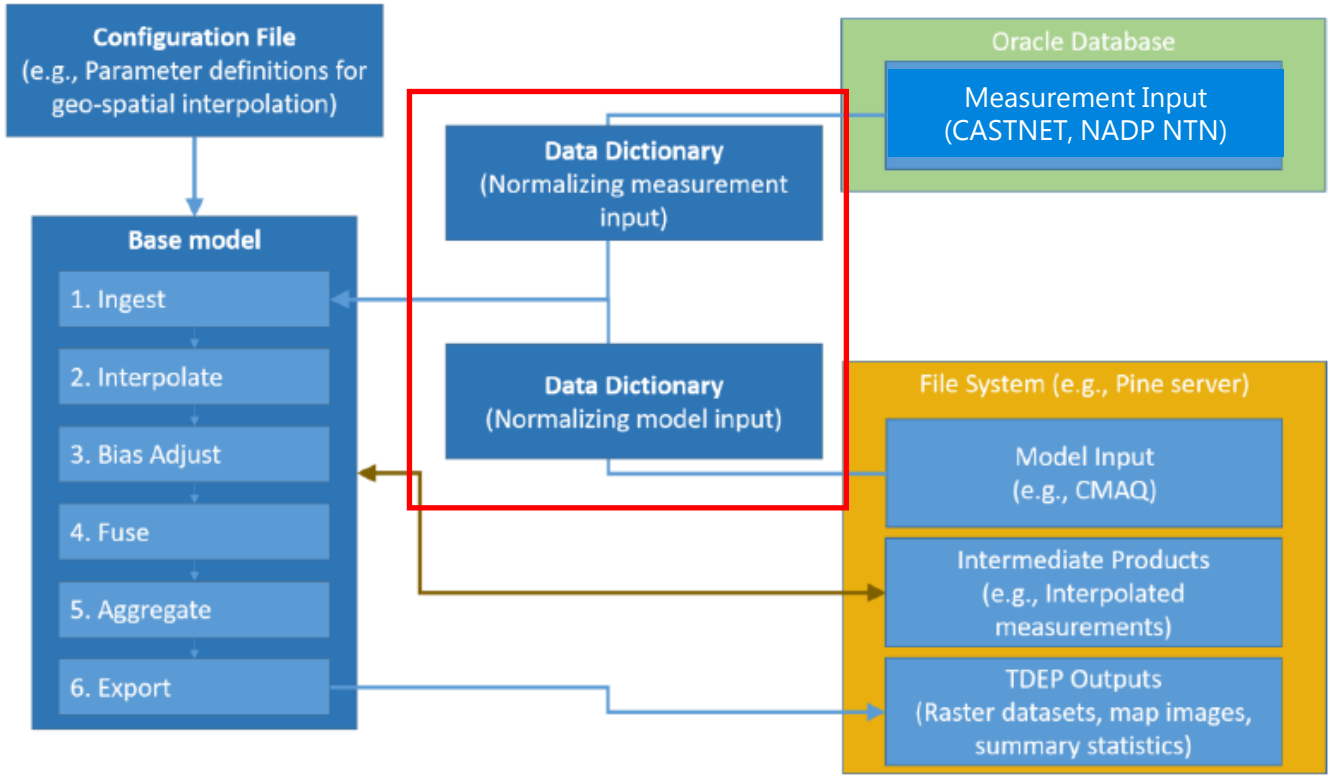
Code Philosophy

- Application functionality split into modules



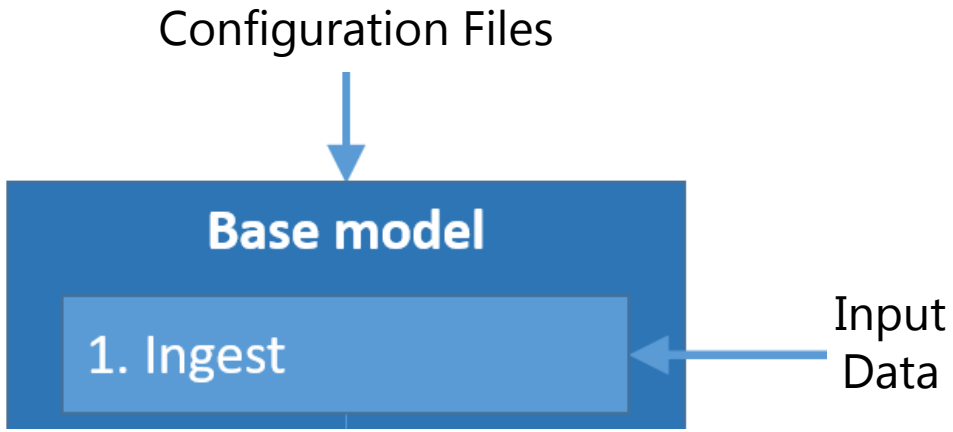
Code Philosophy

- Application functionality split into modules
- Application runs are controlled by configuration files (data dictionaries that control configurations such as years to run, file paths, etc.)



Code Philosophy

- Application functionality split into modules
- Application runs are controlled by configuration files (data dictionaries that control configurations such as years to run, file paths, etc.)
- Operations within Modules are performed by Methods



Ingest Method	Description
cmaq_aggregate_to_ascii	Ingest and format CMAQ NetCDF files
cmaq_ascii2raster	Ingest and format CMAQ NetCDF files
CASTNET_to_point	Ingest and format CASTNET concentration data
NADP_to_point	Ingest and format NADP NTN and AirMon measured precipitation and precipitation chemistry data
prism_ascii2raster	Ingest and format PRISM modeled precipitation data

Ingest Module

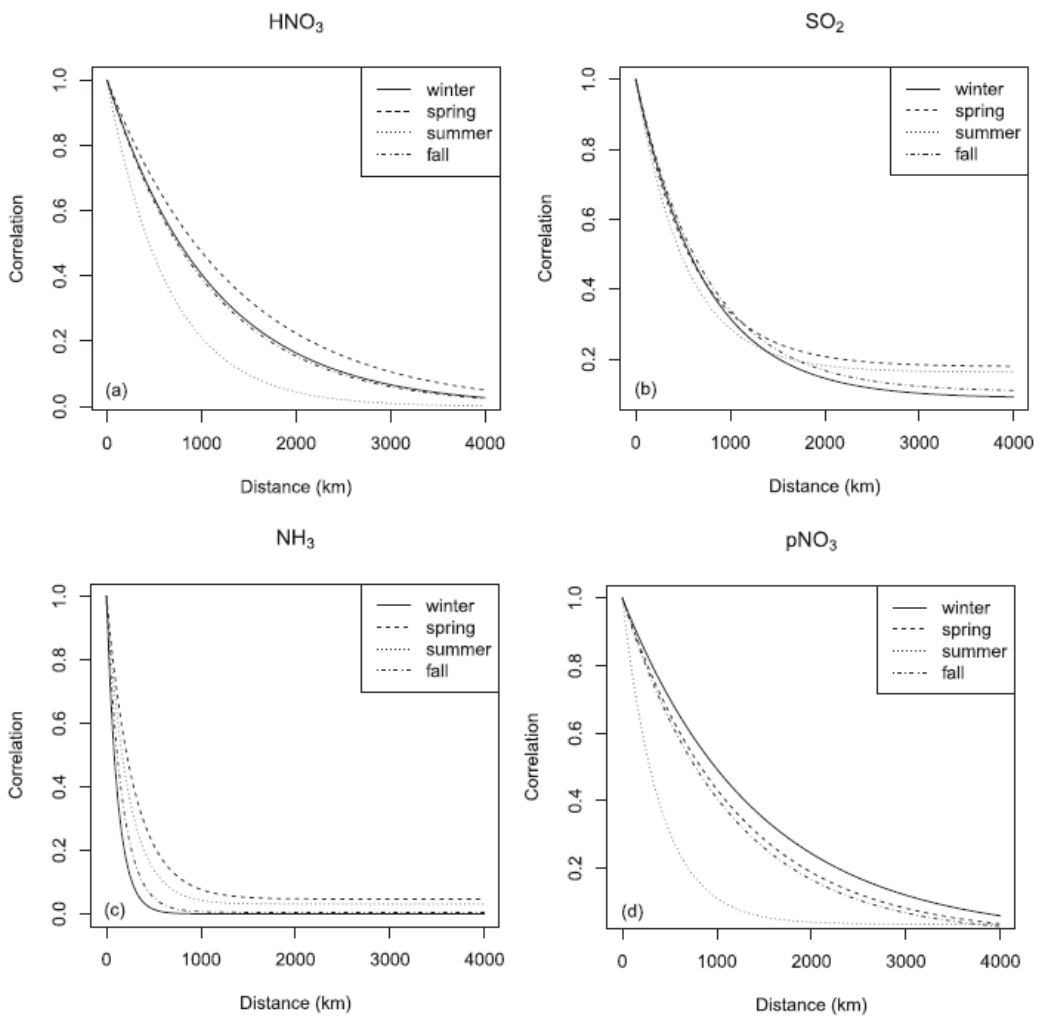
- Contains methods that import data, re-project the data to the desired projection, and export them to:
  - Raster files (i.e., geotiff) for gridded datasets
  - Point shapefiles for point datasets
- The ingested data include
  - CMAQ
  - PRISM
  - CASTNET
  - NADP

```python
class IngestData:
    def __init__(self, work_space_path=config.work_space_path):...

    @staticmethod
    def cmaq_aggregate_to_ascii(year, first_hour, hours_of_average, analyte, config):...

    def cmaq_ascii2raster(self, analyte):
        """
        convert CMAQ ascii file to raster and project to the coordinate system of the final output
        :param: analyte: ddep, aconc, ions, vdwtd, or depv
        :return: CMAQ raster files (.tif)
        """
        out_coor_system = self.data_dict['projection']['output']
        in_coor_system = self.data_dict['projection']['CMAQ']
        input_path = self.data_dict['CMAQ']['CMAQ_ascii_path']
        output_path = os.path.join(self.data_dict['CMAQ']['CMAQ_raster_path'], analyte)
        if not os.path.exists(output_path):
            os.makedirs(output_path)
        input_file_all = glob.glob(os.path.join(input_path, analyte, '*.asc'))
        for input_file in input_file_all:
            output_file = '{}{}'.format(os.path.basename(input_file).split('.')[0], '.tif')
            IngestData.ascii_to_raster(input_file, output_path, output_file, out_coor_system, in_coor_system)

    def prism_ascii2raster(self, year):
        """
        Convert prism ascii file to raster with final product coordinate system
        :param year: model year
        :return: raster file
        """
        out_coor_system = self.data_dict['projection']['output']
        in_coor_system = self.data_dict['projection']['PRISM']
        input_path = self.data_dict['PRISM']['PRISM_path']
        output_path = self.data_dict['PRISM']['PRISM_raster_path']
        input_file = glob.glob(os.path.join(input_path, '*{}*'.format(year)))[0]
        output_file = '{}{}'.format(os.path.basename(input_file).split('.')[0], '.tif')
        IngestData.ascii_to_raster(input_file, output_path, output_file,
                                   out_coor_system, in_coor_system, data_source='prism')
```

Code Example

Interpolation Module

- Contains methods that interpolates the point data
  - CASTNET
    - Ambient concentration
  - NADP
    - Precipitation
    - Precipitation chemistry
- Inverse distance weighting (IDW)
  - *arcpy.sa.Idw()*
  - The interpolation radius for CASTNET was based on analysis of the correlation between data points - the distance where the correlation is 0.7 was used as interpolation radius

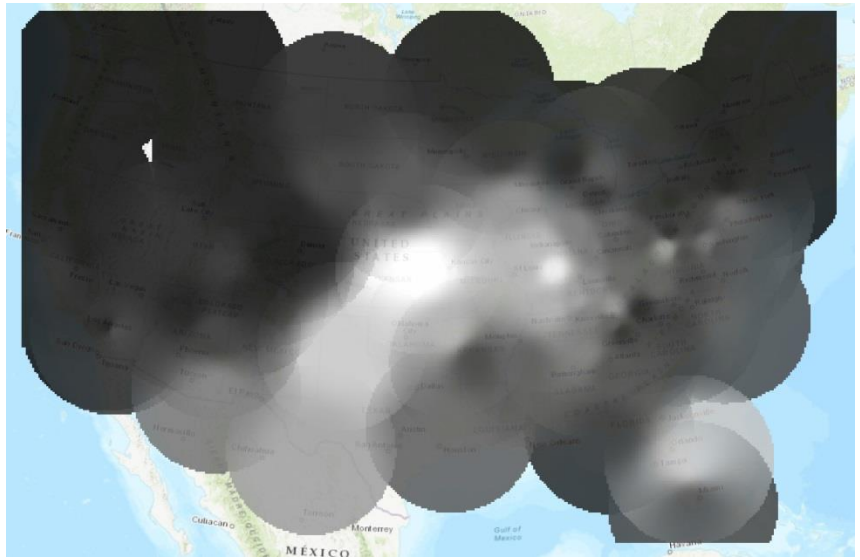


Correlation of the measured concentration by distance

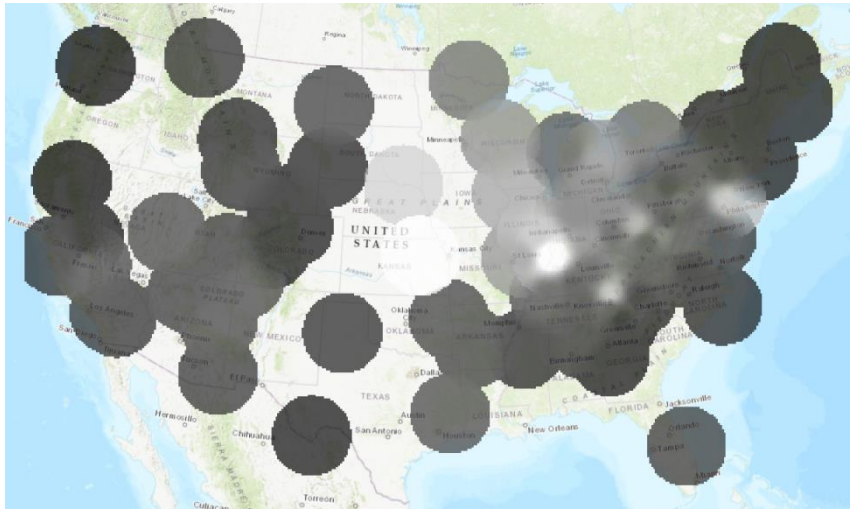
*Schwede and Lear, *Atmospheric Environment*, 2014

Interpolation Module

- Contains methods that interpolates the point data
  - CASTNET
    - Ambient concentration
  - NADP
    - Precipitation
    - Precipitation chemistry
- Inverse distance weighting (IDW)
  - *arcpy.sa.Idw()*
  - The interpolation radius for CASTNET was based on analysis of the correlation between data points - the distance where the correlation is 0.7 was used as interpolation radius



Winter

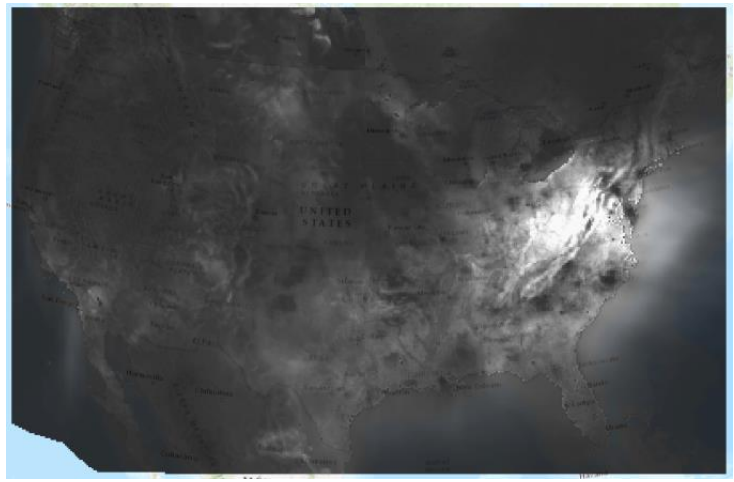


Summer

Bias Adjustment Module

- The bias in CMAQ concentrations is carried over to the CMAQ modeled dry deposition
- Adjustment factors were calculated at each CASTNET sites and interpolated with IDW
- The interpolated bias adjustment factor was applied to the CMAQ dry deposition to correct CMAQ dry deposition bias

Interpolated adjustment factor



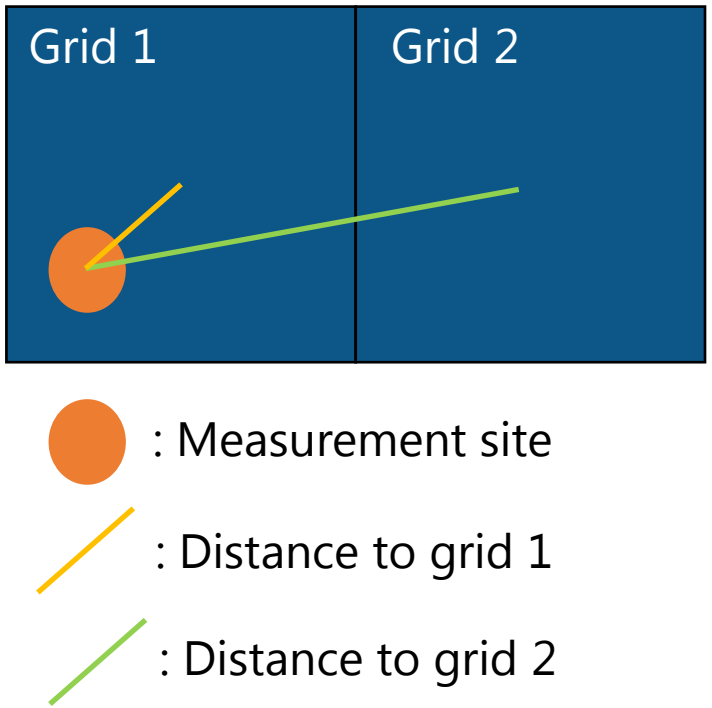
Bias adjusted CMAQ dry deposition

Fuse Module

- A weight factor was assigned for measured and modeled data

  $Q_{fused} = Q_{measured} * W_{measured} + Q_{modeled} * W_{modeled}$

- The weight factors, W, was calculated based on the *distances* from the modeled data grids to the closest measurement site
- The calculation was implemented with *arcpy.sa.EucDistance()*
- The weight was calculated as:
  - $W_{measured} = 1 – (distance/maximum\ radius)$
  - $W_{modeled} = 1 - W_{measured}$
  - The maximum radius was determined by the correlation plot, as mentioned in the interpolation module



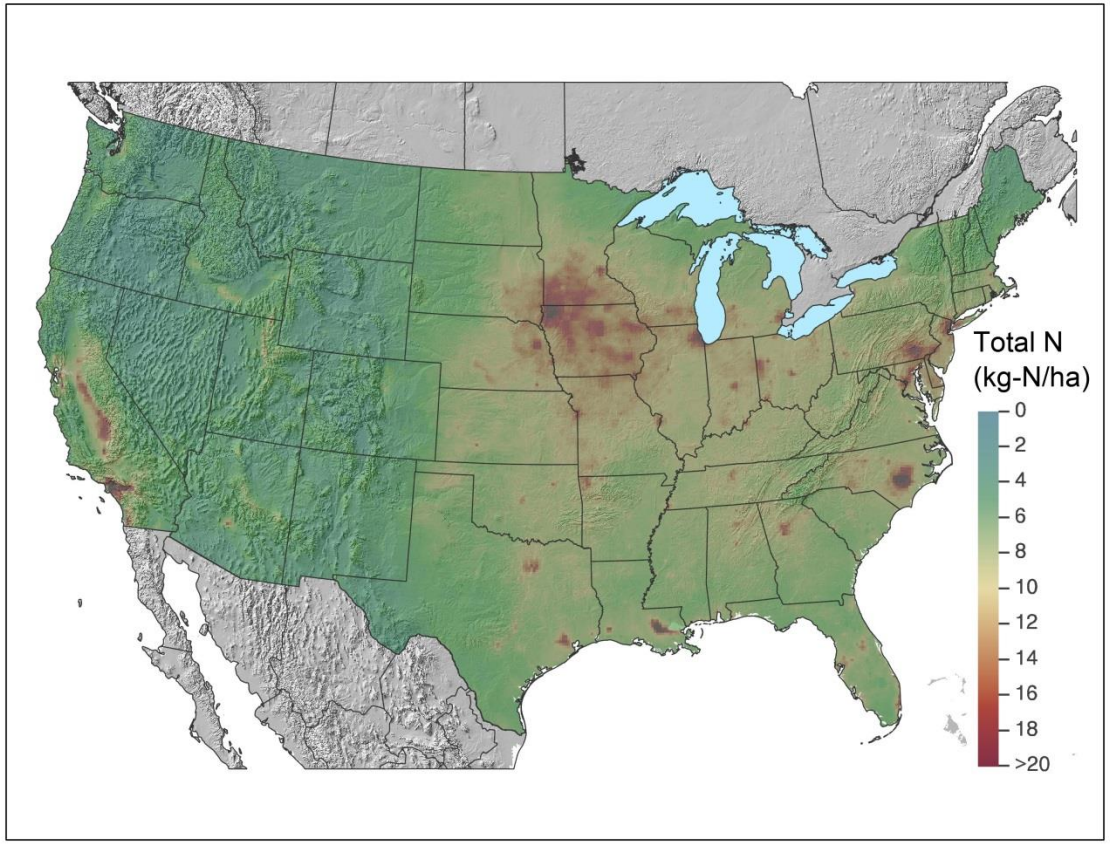
Fuse Module

- Calculate measured dry deposition
  - Interpolated CASTNET * CMAQ deposition velocity
- Dry deposition fusion
  - CMAQ bias-adjusted dry deposition
  - Measured dry deposition
- Precipitation fusion
  - PRISM
  - Interpolated NADP precipitation
- Calculate wet deposition
  - Interpolated NADP precipitation chemistry * fused precipitation

Fused dry deposition

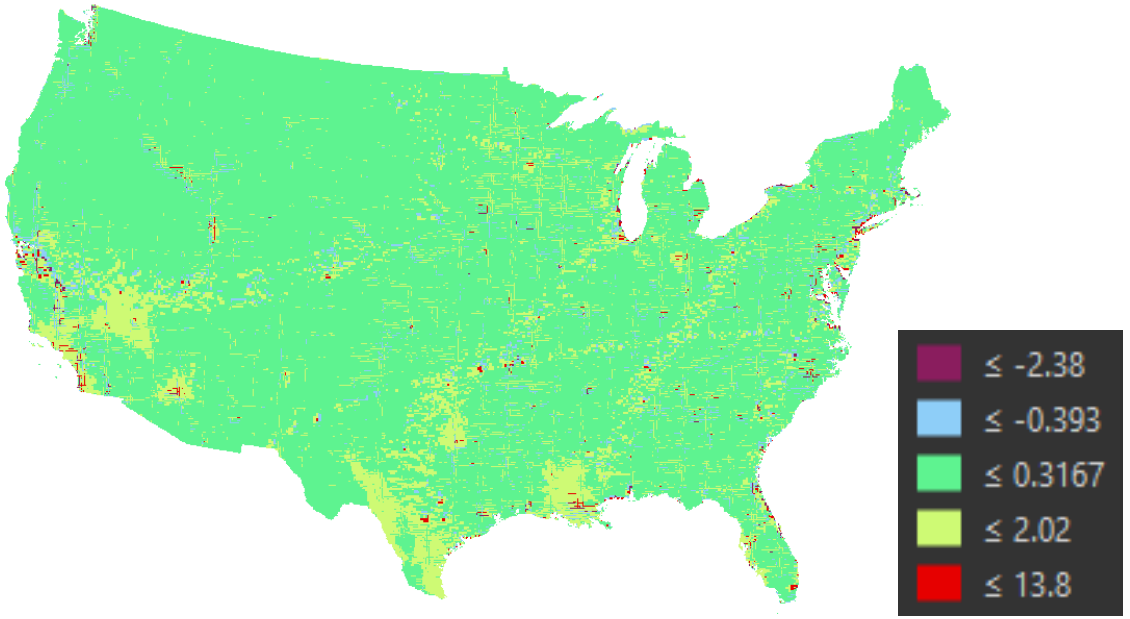
Aggregate and Export Module

- Combine dry and wet deposition to create total deposition raster
- Aggregate the deposition from different species to generate total nitrogen and sulfur deposition
- Export maps, statistics, and raster



Results

- For most of the locations, the Python version agrees well with the AML version
- Differences compared to the AML version:
  - The change from GRS80 projection to the more modern NAD83 datum
  - The change in resolution from 4.13 km to 4 km
  - Slight change in the domain extent
  - Issue with CMAQ projection conversion
  - Unable to reproduce the ion weekly aggregation results
  - Discrepancy in some particulate matter species
- Ongoing effort to fine tune the approach and code, and QA



Difference between the AML version and the current python version for total nitrogen deposition

Results

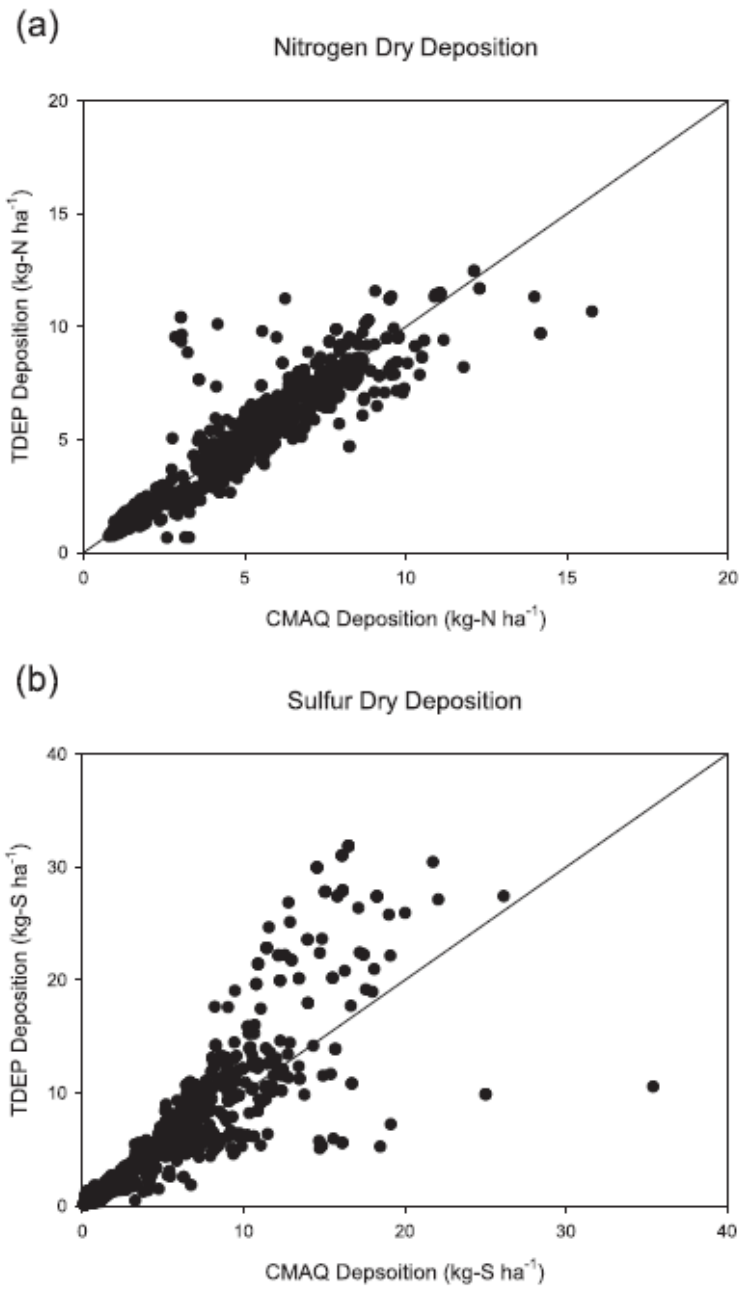
- Reduced the size of the application

Framework	Programming Languages	Number of Scripts	Lines of Code
Current	PERL, SQL, and AML	63	~14,000
Updated	Python	7	~3,200

- User friendly
  - The configuration files make it flexible; users can customize runs without changing the source code
  - Command line options to control certain parameters also available
  - Detailed log files for easy debugging
- Developer friendly
  - Configuration files serve as the interface for input/output to allow for future inclusion of new data
  - Modularized code structure allow developers to add new methods or modules easily for future adaption of new techniques into the code

Results

- For some pollutants, TDep fused dry deposition agrees well with CMAQ dry deposition
- For sulfur dry deposition, some individual sites are substantially higher than CMAQ
  - Western U.S. has few network sites but is with high CMAQ bias
  - Future development should include other networks



*Schwede and Lear, *Atmospheric Environment*, 2014

Summary and Future Improvements

- A user- and developer-friendly implementation of the TDep application was developed with Python geoprocessing
  - Facilitates adaption of future new techniques
- There are differences in the results produced by the Python version compared to the AML version; there are ongoing efforts to identify the cause and debug
- Future improvement:
  - Include other monitoring network for bias adjustment
  - Fuse CMAQ wet deposition with the measured wet deposition

STi | Sonoma Technology

Thank you!

Shih Ying (Changsy) Chang PhD
Air Quality Scientist
cchang@sonomatech.com