



ELSEVIER

Contents lists available at ScienceDirect

MethodsX

journal homepage: [www.elsevier.com/locate/mex](http://www.elsevier.com/locate/mex)

## Method Article

# Weight-of-evidence approach to identify regionally representative sites for air-quality monitoring network: Satellite data-based analysis



Nirav L Lekinwala<sup>1</sup>, Ankur Bharadwaj<sup>3</sup>, Ramya Sunder Raman<sup>3,4,\*</sup>,  
Mani Bhushan<sup>1,2</sup>, Kunal Bali<sup>5</sup>, Sagnik Dey<sup>5,6</sup>

<sup>1</sup> Department of Chemical Engineering, IIT Bombay

<sup>2</sup> Interdisciplinary Programme in Climate Studies, IIT Bombay, Mumbai - 400076, Maharashtra, India

<sup>3</sup> Department of Earth and Environmental Sciences, IISER Bhopal

<sup>4</sup> Center for Research on Environment and Sustainable Technologies, IISER Bhopal, Bhopal Bypass Road, Bhauri, Bhopal - 462 066, Madhya Pradesh, INDIA

<sup>5</sup> Center for Atmospheric Science, IIT Delhi, Delhi - 110016, Delhi, India

<sup>6</sup> Centre of Excellence for Research on Clean Air (CERCA), IIT Delhi

## A B S T R A C T

The methodology discussed in Lekinwala et al., 2020, hereinafter referred to as the 'parent article', is used to setup a nation-wide network for background PM<sub>2.5</sub> measurement at strategic locations, optimally placing sites to obtain maximum regionally representative PM<sub>2.5</sub> concentrations with minimum number of sites. Traditionally, in-situ PM<sub>2.5</sub> measurements are obtained for several potential sites and compared to identify the most regionally representative sites [4], Wongphatarakul et al., 1998) at the location. The 'parent article' proposes the use of satellite-derived proxy for aerosol (Aerosol Optical Depth, AOD) data in the absence of in-situ PM<sub>2.5</sub> measurements. This article focuses on the details about satellite-data processing which forms part of the methodology discussed in the 'parent article'. Following are some relevant aspects:

- High resolution AOD is retrieved from Moderate Resolution Imaging Spectroradiometer (MODIS) instruments aboard NASA's Aqua and Terra satellite using Multi-Angle Implementation of Atmospheric Correction (MAIAC) algorithm. The data is stored as grids of size 1200 × 1200 and a total of seven such grids cover the Indian land mass. These grids were merged, regridded and multiplied by conversion factors from GEOS-Chem Chemical Transport Model to obtain PM<sub>2.5</sub> values. Standard set of tools like CDO and NCL are used to manipulate the satellite-data (\*.nc files).
- The PM<sub>2.5</sub> values are subjected to various statistical analysis using metrics like coefficient of divergence (CoD), Pearson correlation coefficient (PCC) and mutual information (MI).

DOI of original article: [10.1016/j.atmosenv.2020.117544](https://doi.org/10.1016/j.atmosenv.2020.117544)

\* Corresponding Author's.

E-mail address: [ramyasr@iiserb.ac.in](mailto:ramyasr@iiserb.ac.in) (R. Sunder Raman).

<https://doi.org/10.1016/j.mex.2020.100949>

2215-0161/© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license.

(<http://creativecommons.org/licenses/by/4.0/>)

ARTICLE INFO

Method name: Satellite-derived PM<sub>2.5</sub> to establish regional representativeness using statistical metrics (CoD, PCC, MI)

Keywords: MODIS, MAIAC Algorithm, Aerosol Optical Depth, Coefficient of Divergence, Pearson Correlation Coefficient, Mutual Information, Python, CDO

Article history: Received 22 April 2020; Accepted 30 May 2020; Available online 04 June 2020

| Specifications table                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Subject Area                          | Environmental Science                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| More specific subject area            | Use of satellite data in Air Quality Monitoring (AQM)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Method name                           | Satellite-derived PM <sub>2.5</sub> to establish regional representativeness using statistical metrics (CoD, PCC, MI)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Name and reference of original method | The current work focuses on implementation of application of Mutual Information [3] as a metric to capture non-linear relationship in the data. Additionally, metrics like CoD and PCC discussed in [7], is used in the study.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Resource availability                 | AOD data can be downloaded from <a href="https://portal.nccs.nasa.gov/datashare/maiac/DataRelease/China/">https://portal.nccs.nasa.gov/datashare/maiac/DataRelease/China/</a><br>AOD to PM <sub>2.5</sub> Conversion factors are not available online and were procured from Dr. Sagnik Dey and group which is based on the study [8]<br>Other software can be downloaded from<br>1. Python: <a href="https://www.python.org/">https://www.python.org/</a><br>2. CDO: <a href="https://code.mpimet.mpg.de/projects/cdo">https://code.mpimet.mpg.de/projects/cdo</a><br>3. NCL: <a href="https://www.ncl.ucar.edu/">https://www.ncl.ucar.edu/</a> |

Method Details

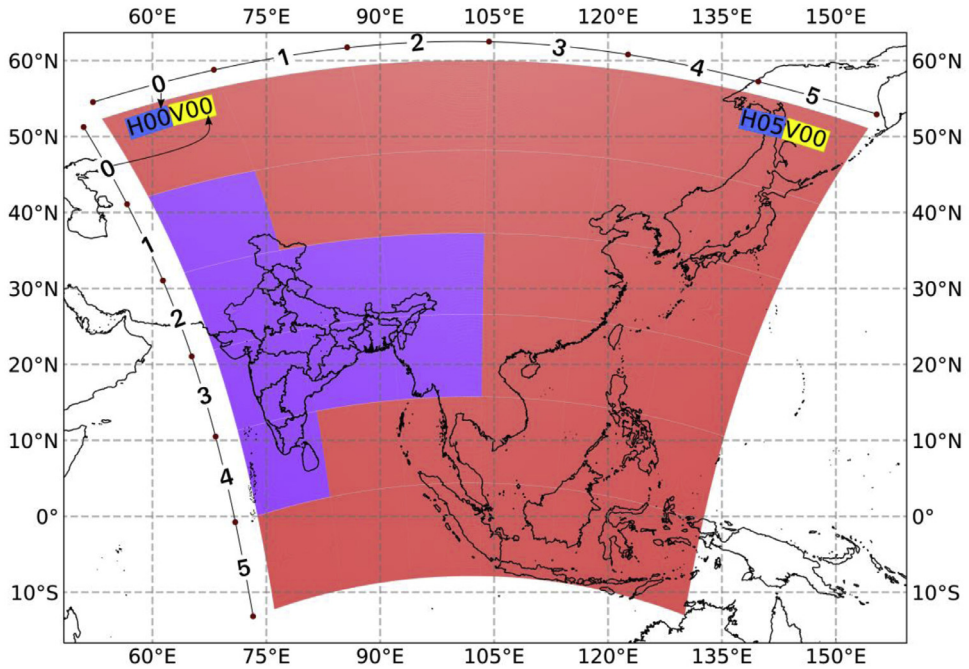
The method discussed in the article covers procurement and pre-processing of the satellite data, analysing them using metrics like Coefficient of Divergence (CoD), Pearson Correlation Coefficient (PCC) and Mutual information (MI) and visualising the spatial map of the metrics. The verification and validation of this method are presented in the parent article [5]. Further, all codes that were developed for this work are presented as supplementary material.

Data Procurement

The high-resolution AOD data is freely available over the Eastern and South-Eastern Asia region and are divided into grids as shown in the Fig. 1 below. Indian landmass is covered by eight grids of size 1200km × 1200 km each as highlighted in Fig. 1.

The files can be downloaded from NCCS's Dataportal [1]. Grid-wise daily files of AOD values are available in the year-wise folders each containing multiple AOT files per day (multiple swaths for Aqua and Terra, max 4 files per day) with unique file names. For example, in grid h00v01 [2], the files are sorted annually; for the year 2004, files are named according to the following convention,

| MAIACAAOT.h00v01.20040010735.hdf |                                                                                                         |
|----------------------------------|---------------------------------------------------------------------------------------------------------|
| Part in Name                     | Description                                                                                             |
| MAIAC                            | Algorithm used on the MODIS data to obtain data products [6]                                            |
| A/T                              | NASA's Aqua/Terra satellite fitted with MODIS sensor                                                    |
| AOT                              | Aerosol optical thickness                                                                               |
| h00v01                           | The grid name as shown in Figure 2                                                                      |
| 2004                             | Year                                                                                                    |
| 001                              | Day of the year, can range from 001 – 365/366 (depending on leap year), 001 is equivalent to January 01 |
| 0735                             | Time at which the satellite passed over the region                                                      |
| hdf                              | File extension                                                                                          |



**Fig. 1.** Scope of the high-resolution data available from MODIS sensor aboard NASA's Aqua and Terra satellite using MAIAC algorithm. The data in purple coloured grids are relevant for this work.

### Data pre-processing

The files for the years 2004 – 2011 were procured for the current methodology as the AOD/AOT conversion factors (CF) were available for these years. Fig. 2 shows the grids over India and the scope of availability of the CF. Fig. 3 schematically shows the data pre-processing steps involved in the methodology.

- 1. Combining the hdf files to netCDF files:** An NCL code to read all the files and extract the variable `Optical_Depth_055` (Aerosol optical depth at 550 nm i.e.  $0.55 \mu\text{m}$ ) and write it to the netCDF file along with latitude, longitude and time data was developed. The code used for combining is attached with the article and is named `processing_01.ncl` which generates `MAIAC_AOT-h00v02-ANNUAL-20040010730-20113650555.nc` file containing data from all hdf files.

```
user@computer$ ncl processing_01.ncl
```

- The details about the newly created netCDF file can be checked using `ncdump` function (as a part of `netcdf-bin` package on a Linux-based PC). `ncdump` of one file is shown below, it shows that there are 9474 time-steps available (multiple daily time instants due to multiple passes of both Aqua and Terra every day).

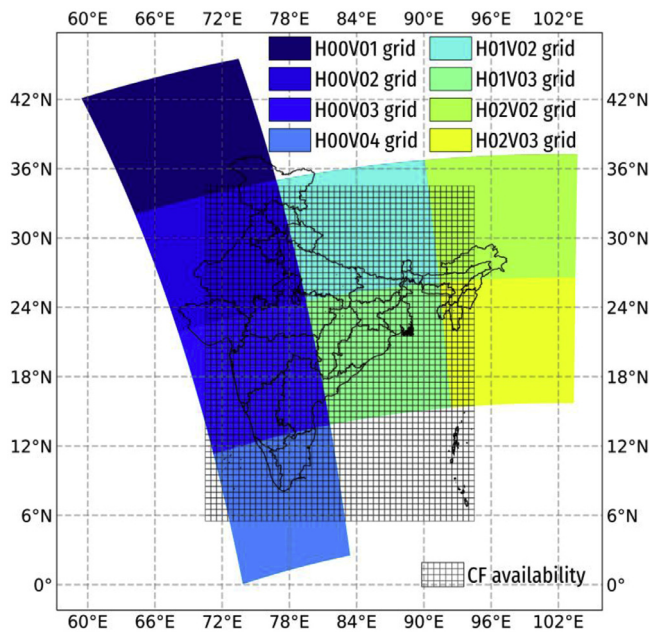


Fig. 2. Scope of 8 grids covering India in different colors and the scope of the available conversion factor shown in mesh.

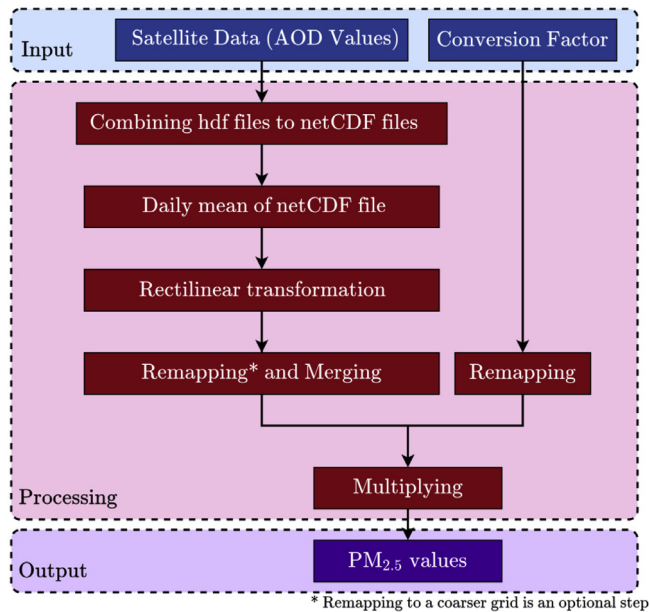


Fig. 3. Schematic of steps involved in data pre-processing

```

user@computer$ ncldump -h MAIACTAOT-h00v02-ANNUAL-20040010730-20113650555.nc # -h options outputs
only the header of the file
netcdf MAIACTAOT-h00v02-ANNUAL-20040010730-20113650555 {
dimensions:
    time = UNLIMITED ; // (9474 currently)
    YDim_grid1km = 1200 ;
    XDim_grid1km = 1200 ;
variables:
    double time(time) ;
        time:units = "hours since 2000-01-01 00:00:00" ;
        time:calendar = "standard" ;
    double lat(YDim_grid1km, XDim_grid1km) ;
        lat:projection = "Albers Conical Equal_Area" ;
        lat:corners = 32.038387901543, 34.9985549202943, 24.5580030404694,
21.8673735745303 ;
        lat:long_name = "latitude" ;
        lat:units = "degrees_north" ;
    double lon(YDim_grid1km, XDim_grid1km) ;
        lon:projection = "Albers Conical Equal_Area" ;
        lon:corners = 64.23634233372, 76.8178669755363, 79.4747290253718,
68.077149355049 ;
        lon:long_name = "longitude" ;
        lon:units = "degrees_east" ;
    short Optical_Depth_055_grid1km(time, YDim_grid1km, XDim_grid1km) ;
        Optical_Depth_055_grid1km:coordinates = "GridLat_grid1km, GridLon_grid1km" ;
        Optical_Depth_055_grid1km:hdfEOS_name="Optical_Depth_055" ;
        Optical_Depth_055_grid1km:projection="Albers Conical Equal_Area" ;
        Optical_Depth_055_grid1km:valid_range=-100s, 5000s ;
        Optical_Depth_055_grid1km:FillValue=-28672s ;
        Optical_Depth_055_grid1km:unit="None" ;
        Optical_Depth_055_grid1km:add_offset=0. ;
        Optical_Depth_055_grid1km:scale_factor=0.001 ;
        Optical_Depth_055_grid1km:long_name="AOT at 0.55 micron" ;

// global attributes:
        :creation_date = "Tue Apr 23 10:18:39 IST 2019" ;
        :Conventions = "None" ;
        :NCL_version = "Any version of NCL >= 5.2.0 (4/2010)" ;
        :source_file = "MAIACTAOT HDF-EOS files" ;
        :title = "Created for COALESCE Project -- time dimension added to selected
variables, created by Kunal Bali, and refined by Nirav L. Lekinwala" ;
}

```

b. The process of converting hdf files into netCDF files is repeated for all the grids. Bottle-neck in this process is the disk writing speed (file for each grid is about 32GB in size) which may take up to an hour per grid. For all the computations, Linux-based computer with 6 core – 12 thread i7-8700 processor, 24 GB RAM and a standard spinning hard-disk drive is used.

2. **Daily mean of netCDF file:** To make the processing easier and bring uniformity across different grids, the daily mean of the data is used, thus making total number of time instants (2922 days) same across different grids.

```

user@computer$ cdo daymean MAIACTAOT-h00v02-ANNUAL-20040010730-20113650555.nc H00V02_ALL.nc
#cdo daymean input.nc output.nc
user@computer$ ncdump -h H00V02_ALL.nc
## ncdump output ##
netcdf H00V02_ALL.nc {
dimensions:
    time = UNLIMITED ; // (2922 currently)
    bnds = 2 ;
    XDim_grid1km = 1200 ;
    YDim_grid1km = 1200 ;
variables:
    double time(time) ;
        time:standard_name = "time" ;
        time:bounds = "time_bnds" ;
        time:units = "hours since 2000-01-01 00:00:00" ;
        time:calendar = "standard" ;
        time:axis = "T" ;
    double time_bnds(time, bnds) ;
    double lat(YDim_grid1km, XDim_grid1km) ;
        lat:long_name = "latitude" ;
        lat:units = "degrees_north" ;
        lat:projection = "Albers Conical Equal_Area" ;
        lat:corners = 32.038387901543, 34.9985549202943, 24.5580030404694,
21.8673735745303 ;
    double lon(YDim_grid1km, XDim_grid1km) ;
        lon:long_name = "longitude" ;
        lon:units = "degrees_east" ;
        lon:projection = "Albers Conical Equal_Area" ;
        lon:corners = 64.23634233372, 76.8178669755363, 79.4747290253718,
68.077149355049 ;
    short Optical_Depth_055_grid1km(time, YDim_grid1km, XDim_grid1km) ;
        Optical_Depth_055_grid1km:long_name = "AOT at 0.55 micron" ;
        Optical_Depth_055_grid1km:add_offset = 0. ;
        Optical_Depth_055_grid1km:scale_factor = 0.001 ;
        Optical_Depth_055_grid1km:_FillValue = -28672s ;
        Optical_Depth_055_grid1km:missing_value = -28672s ;
        Optical_Depth_055_grid1km:hdfeos_name = "Optical_Depth_055" ;
        Optical_Depth_055_grid1km:projection = "Albers Conical Equal_Area" ;
        Optical_Depth_055_grid1km:unit = "None" ;

// global attributes:
    :CDI = "Climate Data Interface version ?? (http://mpimet.mpg.de/cdi)";
    :Conventions = "None" ;
    :history = "Tue Apr 23 14:10:09 2019: cdo -P 6 daymean MAIACTAOT-h00v02-ANNUAL-
20040010730-20113650555.nc H00V02_ALL.nc" ;
    :creation_date = "Tue Apr 23 10:19:00 IST 2019" ;
    :NCL_version = "Any version of NCL >= 5.2.0 (4/2010)" ;
    :source_file = "MAIACTAOT HDF-EOS files" ;
    :title = "Created for COALESCE Project -- time dimension added to selected
variables, created by Kunal Bali, and refined by Nirav L. Leginwala" ;
    :frequency = "day" ;
    :CDO = "Climate Data Operators version 1.9.3 (http://mpimet.mpg.de/cdo)" ;
    :cdo_openmp_thread_number = 6 ;
}

```

3. **Rectilinear transformation:** The data grids obtained from the satellite are curvilinear in nature. The daily-averaged data for different grids needs to be merged to create a consistent spatio-temporal dataset over all the time instants. In order to merge the grids, they need to be transformed from the curvilinear form to rectilinear form using code `rectilinear.ncl` which is later remapped using grid characteristics given in `regrid.txt`.
  - a. The NCL code creates a blank rectilinear \*.nc file.
  - b. The blank rectilinear \*.nc file is populated with the variables and saved separately using CDO's `setgrid` function.

```
## Rectilinear transformation - Creating blank netCDF file
user@computer$ ncl H00V02_ALL_rectilinear.ncl ## A file named H00V02_ALL_rect.nc is created.

## Rectilinear transformation - Populating the blank netCDF file with variables
user@computer$ cdo -P 12 -f nc -setgrid,H00V02_ALL_rect.nc H00V02_ALL.nc H00V02_ALL_regrid.nc
## Description
cdo      : is the climate data operator software package
-P 12    : Option that enables using multiple processors to speed up the process, 12 are the number
of processors used for the task.
-f nc    : Option to save the file in netCDF format
-setgrid,H00V02_ALL_rect.nc : The grid needs to be created with the specifications of
H00V02_ALL_rect.nc, copy the data from file H00V02_ALL.nc and save it as H00V02_ALL_regrid.nc
##
```

4. **Remapping and Merging:** The merging of grid is important to ensure spatial continuity in the data and makes processing easy. Following are some points to note,
  - a. To merge the different grids, a large grid covering the extent of the smaller grids needs to be created.
  - b. Remapping process is computationally expensive and time consuming and thus to establish a balance between spatial resolution and computation time, for the current application, the 1 km × 1 km data was remapped (converted) to 1.5 km × 1.5 km using CDO's `remapcon` function.
  - c. This resolution was found optimal as without losing fineness in the data, the computation time was reduced by about 2-2.5 times (about half the original time).
  - d. It took about 400 hours for the Remapping to complete on a 6-core, 12-thread 4GHz Intel i7-8700 machine with 24GB of RAM.
  - e. Remapping the data to the original spatial resolution is essential before merging, while Remapping it to a coarser grid is optional.
  - f. During the merging process (using CDO's `mergegrid` function), a minimum 500 GB of storage is required for all the auxiliary files (intermediate files) which can be deleted after successful merging.
  - g. In case of some issue or some error in the file, the computation can be resumed from a step before the error occurred if the auxiliary files are saved until successful completion of Remapping and merging.

```
## Remapping and Merging - Remapping the data to a coarser resolution
user@computer$ cdo -P 12 -remapycon,regrid.txt H00V02_ALL_regrid.nc H00V02_ALL_rectlin.nc

## Remapping and Merging - Merging the grids one by one
# Merging H00V01 with H00V02 as temp_02.nc
user@computer$ cdo -P 12 mergegrid H00V01_ALL_rectlin.nc H00V02_ALL_rectlin.nc temp_01.nc
user@computer$ cdo -P 12 mergegrid temp_01.nc H00V02_ALL_rectlin.nc temp_02.nc
# Merging temp_02 grids with H00V03 as temp_04.nc
user@computer$ cdo -P 12 mergegrid temp_02.nc H00V03_ALL_rectlin.nc temp_03.nc
user@computer$ cdo -P 12 mergegrid temp_03.nc H00V03_ALL_rectlin.nc temp_04.nc
# Merging temp_04 grids with H00V04 as temp_06.nc
user@computer$ cdo -P 12 mergegrid temp_04.nc H00V04_ALL_rectlin.nc temp_05.nc
user@computer$ cdo -P 12 mergegrid temp_05.nc H00V04_ALL_rectlin.nc temp_06.nc
# Merging temp_06 grids with H01V02 as temp_08.nc
user@computer$ cdo -P 12 mergegrid temp_06.nc H01V02_ALL_rectlin.nc temp_07.nc
user@computer$ cdo -P 12 mergegrid temp_07.nc H01V02_ALL_rectlin.nc temp_08.nc
# Merging temp_08 grids with H01V03 as temp_10.nc
user@computer$ cdo -P 12 mergegrid temp_08.nc H01V03_ALL_rectlin.nc temp_09.nc
user@computer$ cdo -P 12 mergegrid temp_09.nc H01V03_ALL_rectlin.nc temp_10.nc# Merging temp_10
grids with H02V02 as temp_12.nc
user@computer$ cdo -P 12 mergegrid temp_10.nc H02V02_ALL_rectlin.nc temp_11.nc
user@computer$ cdo -P 12 mergegrid temp_11.nc H02V02_ALL_rectlin.nc temp_12.nc
# Merging temp_12 grids with H02V03 as ALLGRID_AOD.nc
user@computer$ cdo -P 12 mergegrid temp_12.nc H02V03_ALL_rectlin.nc temp_13.nc
user@computer$ cdo -P 12 mergegrid temp_13.nc H02V03_ALL_rectlin.nc ALLGRID_AOD.nc
```

5. **Remapping CF and Multiplying AOD with CF:** For the current study, the CF are obtained from the GEOS-Chem Chemical transport model which is adopted from the study of van Donkelaar et al., [8].
- The daily CF (2004 - 2011) are already in rectilinear form, it is easy to remap it to same spatial resolution as the AOD.
  - The daily CF value is multiplied with the daily AOD values to obtain daily PM<sub>2.5</sub> values.
  - The multiplication is computationally expensive and is also bottlenecked by the storage speed, it may take about 4-6 hours to process about  $15.4 \times 10^{12}$  floating point operations.

```
## Remapping the Conversion factor and Multiplying
user@computer$ cdo -P 12 -remapycon,regrid.txt CF_2004.nc CF_2004_regrid.nc

## Merging all the regridded conversion factors
user@computer$ cdo mergetime CF_2004_regrid.nc <-> CF_2011_regrid.nc CF_ALL.nc # CF_ALL.nc is the
merged file of all the conversion factors over 2004 - 2011

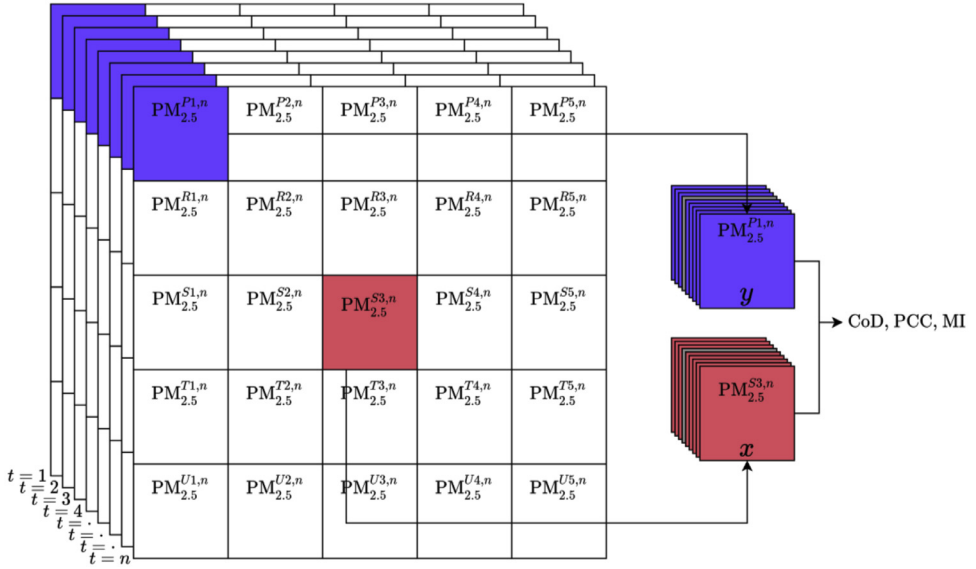
## Obtaining the PM values by multiplying AOD and CF values.
user@computer$ cdo -b 16 -P 12 -f nc -mul -selname,Optical_Depth_Q55_grid1km ALLGRID_AOD.nc -
selname,var1 CF_ALL.nc PM_ALL.nc
```

## Data Analysis

The satellite-derived PM<sub>2.5</sub> values obtained in the data pre-processing step is used for further analysis. A schematic of the PM<sub>2.5</sub> data obtained is shown in Fig. 4.

Following are the important points to note in data analysis,

1. A python code namely computation.py was developed to compute the different metrics like coefficient of divergence (CoD), Pearson correlation coefficient (PCC) and mutual information (MI).
2. The reference cell (location where the sampler is planned to be placed) is identified in the data using site's latitude and longitude. For all the cells in a region of about 300 km around the site, values of the metrics are calculated sequentially when the values are available at same time instants.



**Fig. 4.** Schematic of arrangement of data and computation of the different metrics

3. Coefficient of Divergence (CoD) is one of the statistical techniques used to compare two values, if the values were close, the metric CoD will have value close to zero, while for completely different values, the CoD will be close to unity. The CoD for two columns of values can be calculated as,

$$CoD = \sqrt{\frac{1}{n} \sum_{t=1}^n \left( \frac{x_t - y_t}{x_t + y_t} \right)^2} \quad (1)$$

where,  $x_t$  and  $y_t$  value of  $PM_{2.5}$  at the reference cell and a neighbouring cell at  $t^{\text{th}}$  time instant respectively,  $n$  are the total number of time instants

A function for CoD was created for all the calculations and is defined as `cod_calculations` in the file `computation.py`

```
# Function for Coefficient of Divergence function
def cod_calculation(x, y):
    calc1 = ((x-y)/(x+y))*2
    cod_val = np.sqrt((1/x.shape[0])*np.sum(calc1))
    return cod_val
```

4. Pearson correlation coefficient (PCC) is an addition statistical metric which is used. It can easily be accessed from Python NumPy library's `corrcoef` function and is used as follows,

```
# Function for Pearson Correlation Coefficient
def pcc_calculation(x, y):
    pcc_val = np.corrcoef(x, y)[0, 1] #np is alias for numpy
    return pcc_val
```

5. While CoD compares the values, and PCC quantifies the linear relationship in the data, mutual information (MI) is used to additionally capture the non-linear relationship in the data.
6. Leginwala et al. [5] discuss the algorithm used to calculate mutual information in detail. Mutual information function `mutual_information` function is part of the `computation.py` code.

a. The mutual information can be computed as follows,

$$I(\tilde{X}, \tilde{Y}) = \sum_{j=1}^N \sum_{i=1}^N P(\tilde{x}_i, \tilde{y}_j) \log \left[ \frac{P(\tilde{x}_i, \tilde{y}_j)}{P(\tilde{x}_i)P(\tilde{y}_j)} \right] \Delta x \Delta y$$

- b. It requires the joint density ( $P(\tilde{x}, \tilde{y})$ ) and marginal density ( $P(\tilde{x})$  and  $P(\tilde{y})$ ) of the data under consideration to compute MI value.
- c. A Gaussian kernel is fit through the  $x, y$  data using `gaussian_kde` function in SciPy library's `stats` class. It can be called using `stats.gaussian_kde(x, y)`.
- d. The fitted joint density function obtained is evaluated at 100 equispaced values in range  $\tilde{x} \rightarrow (\mu_{\tilde{x}} - 6\sigma_{\tilde{x}}) : \Delta x : (\mu_{\tilde{x}} + 6\sigma_{\tilde{x}})$  and  $\tilde{y} \rightarrow (\mu_{\tilde{y}} - 6\sigma_{\tilde{y}}) : \Delta y : (\mu_{\tilde{y}} + 6\sigma_{\tilde{y}})$  which is then summed across all the values of  $\tilde{x}$  to obtain  $P(\tilde{y})$  and vice-versa.
- e. The complete `mutual_information` function with appropriate comments about the lines of codes is given below,

```
# Function for mutual information
def mutual_information(x, y, discrete, discrete_inverse):
    gridx = np.linspace(min(x)-(6*np.std(x)), max(x)+(6*np.std(x)), discrete) #min-6sigma and
    max+6sigma for x data
    gridy = np.linspace(min(y)-(6*np.std(y)), max(y)+(6*np.std(y)), discrete) #min-6sigma and
    max+6sigma for y data

    values = np.vstack([x, y]) #stacking the arrays one below other
    fxy = stats.kde.gaussian_kde(values) #2-D KDE using scott's bandwidth factor
    fxy.set_bandwidth(bw_method='silverman')

    delx = (gridx[discrete-1]-gridx[0])*discrete_inverse #dx - used in integration
    dely = (gridy[discrete-1]-gridy[0])*discrete_inverse #dy - used in integration
    gx, gy = np.meshgrid(gridx, gridy)

    xy_joint = fxy.evaluate([list(gx.ravel('F'))], list(gy.ravel('F'))])
    xy_joint = xy_joint.reshape((discrete, discrete)).squeeze()

    x_marginal = np.sum(xy_joint, axis=1)*delx #marginal distribution of x
    y_marginal = np.sum(xy_joint, axis=0)*dely #marginal distribution of y

    [X, Y] = np.meshgrid(x_marginal, y_marginal) #meshgrid for calculation
    xy_joint_list = xy_joint.ravel('F')
    x_marginal_list = X.ravel('F')
    y_marginal_list = Y.ravel('F')

    #mutual information calculation
    calculation = ((np.log(xy_joint_list)-np.log(x_marginal_list)-
    np.log(y_marginal_list))*xy_joint_list)*delx*dely

    #removing the nan values
    calculation = calculation[np.isfinite(calculation)]

    #summing to get the single MI value
    calculation = np.sum(calculation)

    #returns the mutual information for x and y
    return calculation
```

7. The calculated values are saved as \*.csv files and can be plotted to obtain spatial plots using Python code which are part of the attached `computation.py` file

## Data Visualisation

The values of CoD, PCC and MI calculated for different sites need to be visualised to interpret their spatial characteristics. Matplotlib is a widely used plotting library in Python. Following are some important steps in plotting,

1. The saved csv file is read using Python's Pandas library using `read_csv` function.
2. Matplotlib's `pcolormesh` function is used to create a colormesh plot based on the values of metric. A `cubehelix` colormap is used for all the plots.
3. Additionally, contours are added to the plot to create a clear demarcation of different values. Matplotlib's `contour` function is used to create it.
4. Several fonts related and contour plot related options are used to make the plot visually better. Other functions and options used in the code are shown below.
5. The plots for MI (a) CoD (b) and PCC (c) created using the code in the `computation.py` file are presented in Fig. 5. Results and interpretation of Fig. 5 for Bhopal site and other sites are discussed in [5].

```

def plotting_function(data, lat, lon, range_value, slocation, metric_type):
    """
    The function generates plots for different metrics used in the study.
    """

    # Metric name and other parameters for visualisation
    if metric_type == 'mi':
        metric_name = 'mutual_information'
        colormap = 'cubehelix_r'
        contour_col = 'white'
    elif metric_type == 'cod':
        metric_name = 'coefficient_of_divergence'
        colormap = 'cubehelix_r'
        contour_col = 'black'
    elif metric_type == 'pcc':
        metric_name = 'pearson_correlation_coefficient'
        colormap = 'cubehelix'
        contour_col = 'white'
    else:
        raise ValueError('Please check the metric type, it can only take values "mi", "cod" or "pcc"')

    # Site location
    r_lon, r_lat = slocation[0], slocation[1]
    site_lon, site_lat = lon[r_lon], lat[r_lat]

    # Truncating the latitude and longitude arrays to a small region
    lat_update = lat[r_lat-range_value:r_lat+range_value]
    lon_update = lon[r_lon-range_value:r_lon+range_value]

    # Generating the figure
    plt.figure(figsize=(20, 20))

    # Setting extent of the map
    ax = plt.axes(projection=ccrs.PlateCarree())
    ax.set_extent([lon[r_lon-range_value-25], lon[r_lon+range_value+25], lat[r_lat-range_value-25], lat[r_lat+range_value+25]])
    range_dictionary = {'mutual_information': [0, 2], 'coefficient_of_divergence': [0, 0.5], 'pearson_correlation_coefficient': [-1, 1]}

```

```

# Plotting the values
cmap = plt.pcolormesh(lon_update, lat_update, data, vmin=range_dictionary[metric_name][0], vmax
= range_dictionary[metric_name][1], cmap=colormap)
cont = plt.contour(lon_update, lat_update, gaussian_filter(data, 3), colors=contour_col)
ax.clabel(cont, inline=3, fontsize=14)
plt.colorbar(cmap, orientation='vertical', shrink=0.6, pad=0.08, format='% .2f')

plt.scatter(site_lon, site_lat, s=50, c='white', transform=ccrs.PlateCarree())
fname = r'/path/to/shape_file/IND_adm/INDIA.shp'

ax.add_geometries(Reader(fname).geometries(), ccrs.PlateCarree(), facecolor='None', edgecolor='gray', lw=1)

gl = ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=True, linewidth=2, color='gray', alpha=0.5, linestyle='--')
gl.xlabel_right = False
gl.ylabel_top = False
gl.xformatter = LONGITUDE_FORMATTER
gl.yformatter = LATITUDE_FORMATTER

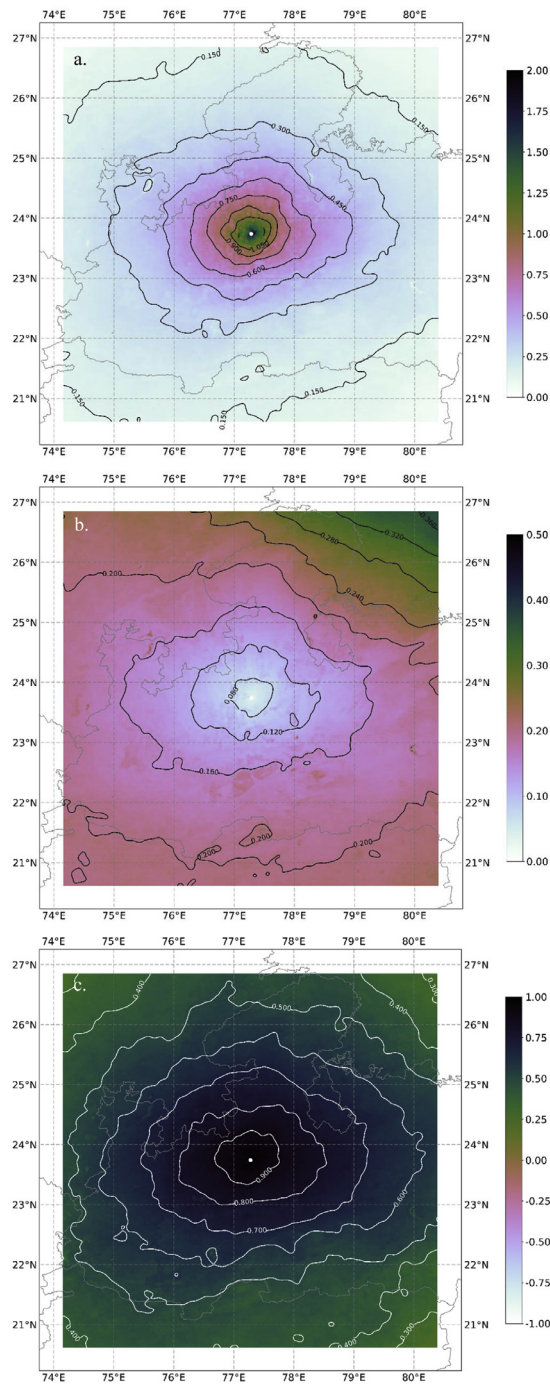
plt.savefig(save_path+inst+'_new_'+metric_name+'.png', bbox_inches='tight', dpi=300)
print('Plot for '+metric_name+' generated')

# Generating figures for different metrics
mi_values = np.array(pd.read_csv(save_path+inst+'_pm_mi_values.csv', delimiter=',', header=None))
plotting_function(mi_values, lat=lat, lon=lon, range_value=range_value, slocation=[r_lon, r_lat], metric_type='mi')

cod_values = np.array(pd.read_csv(save_path+inst+'_pm_cod_values.csv', delimiter=',', header=None))
plotting_function(cod_values, lat=lat, lon=lon, range_value=range_value, slocation=[r_lon, r_lat], metric_type='cod')

r2_values = np.array(pd.read_csv(save_path+inst+'_pm_r2_values.csv', delimiter=',', header=None))
plotting_function(r2_values, lat=lat, lon=lon, range_value=range_value, slocation=[r_lon, r_lat], metric_type='pcc')

```



**Fig. 5.** Figures generated using the aforementioned code for (a) Mutual Information, (b) Coefficient of Divergence and (c) Pearson correlation coefficient

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by the MOEFCC under the NCAP-COALESCe project {Grant No.14/10/2014-CC(Vol.II)}. The authors thank the internal review committee of the NCAP-COALESCe project for their comments and suggestions on this paper. The views expressed in this document are solely those of authors and do not necessarily reflect those of the Ministry. The Ministry does not endorse any products or commercial services mentioned in this publication.

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.mex.2020.100949](https://doi.org/10.1016/j.mex.2020.100949).

## References

- [1] NCCS Dataportal – Datashare, 1 (last accessed on January 03, 2020) available at <https://portal.nccs.nasa.gov/datashare/maiac/DataRelease/China/>.
- [2] NCCS Dataportal – Datashare, 2 (last accessed on January 03, 2020) available at <https://portal.nccs.nasa.gov/datashare/maiac/DataRelease/China/h00v01/2004/>.
- [3] Cover, T.M.; Thomas, J.A. (1991). Elements of Information Theory (Wiley ed.). ISBN 978-0-471-24195-9.
- [4] E. Kim, P.K. Hopke, J.P. Pinto, W.E. Wilson, Spatial variability of fine particle mass, components, and source contributions during the Regional Air Pollution Study in St. Louis. Environ. Sci. Technol. 39 (2005) 4172–4179 <https://doi.org/10.1021/es049824x>.
- [5] Leginwala, L. N., Bhardwaj, A., Sunder Raman, R., Bhushan, M., Bali, K., Dey, S., 2020. A framework for setting up a country-wide network of regional surface PM<sub>2.5</sub> sampling sites utilising a satellite-derived proxy – The COALESCe project, India. Atmos. Environ. <https://doi.org/10.1016/j.atmosenv.2020.117544>.
- [6] A. Lyapustin, Y. Wang, I. Laszlo, R. Kahn, S. Korkin, L. Remer, R. Levy, J.S. Reid, Multiangle implementation of atmospheric correction (MAIAC): 2. Aerosol algorithm, J. Geophys. Res. 116 (2011) D03211 <https://doi.org/10.1029/2010JD014986>.
- [7] V. Wongphatarakul, S.K. Friedlander, J.P. Pinto, A comparative study of PM<sub>2.5</sub> ambient aerosol chemical databases, Environ. Sci. Technol. 32 (1998) 3926–3934 <https://doi.org/10.1021/es9800582>.
- [8] A. van Donkelaar, R.V. Martin, M. Brauer, R. Kahn, R. Levy, C. Verduzco, P.J. Villeneuve, Global estimates of ambient fine particulate matter concentrations from satellite-based aerosol optical depth: development and application, Environmental health perspectives 118 (6) (2010) 847–855 <https://doi.org/10.1289/ehp.0901623>.