Pollution Detection Algorithm (PDA) code description

Author: Ivo Beck (ivo.beck@epfl.ch)

Release: 2021-12-06

For a detailed scientific description see Beck et al. (submitted to AMT).

About the Pollution Detection Algorithm (PDA)

The PDA was developed to identify pollution from local pollution sources in aerosol and trace gas datasets obtained at remote locations to be able to remove data, which are not representative of that particular environment. For the development we used data collected during the MOSAiC (Multidisciplinary drifting Observatory for the Study of Arctic Climate) expedition in the central Arctic from September 2019 to October 2020. A more detailed description of the expedition is described in (Shupe et al., under review). The PDA identifies and flags periods of polluted data in five steps. The first and most important step identifies polluted periods based on the gradient (time-derivative) of a concentration over time (Steps 1A and 1B). If this gradient exceeds a given threshold, data are flagged as polluted. Further pollution identification steps are a simple concentration threshold filter (Step 2), a neighboring points filter (Step 3, optional), a median (Step 4) and a sparse data filter (Step 5, optional). The PDA only relies on the target dataset itself and is independent of ancillary datasets such as meteorological variables. All parameters of each step are adjustable so that the PDA can be "tuned" to be more or less stringent (e.g., flag more or less data points as polluted). In order to use the PDA, we recommend to read the detailed description of Beck et al. (submitted to AMT).

How to run the code with Python

The code runs from the command line (terminal) on Windows, Linux or Mac. All you need is to have python installed on your computer. No GUI is needed. The script will ask you for the path to the data and for the path to save the plots which are needed to make decisions regarding different filtering steps and the final dataset with a pollution flag.

1. Install **python 3** on your computer.
   - Follow this link to download python: https://www.python.org/downloads/.
2. Install the following packages and modules (or make sure they are already installed).

2.a. If not installed, you need to install **pip**, a package-management system written in Python used to install and manage software packages.

- For windows, follow these instructions: https://phoenixnap.com/kb/install-pip-windows
- For mac users, follow these instructions: https://phoenixnap.com/kb/install-pip-mac

2.b. If not installed already, install **NumPy**, a library adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. This can be done in the terminal, using the pip command:

- *pip install numpy*

2.c. If not installed already, install **pandas**, a library for data manipulation and analysis.

2.d. If not installed already, install the Python mathematics library **Matplotlib**.

2.e. If not installed already, install **seaborn**, a data visualization library based on matplotlib.

3. Create a folder for the PDA script, data and plots.
4. Open “PDA_script.py” using the command prompt.
   - To start python with the command prompt in Windows: https://www.wikihow.com/Use-Windows-Command-Prompt-to-Run-a-Python-File
   - To start python with the terminal on mac: https://www.maketecheasier.com/run-python-script-in-mac/
   - Change the directory in the command line to the folder of the script: *cd {yourpath}/PDA_script*
   - Start python script: *python3 PDA_script.py*

Dataset preparation before running the script

The dataset should be a comma separated text file with only two columns, first a date+time column and second a concentration column. The first line of the file is expected to be the header (but this can also be left empty). Header names do not have specific requirements, since they will be re-named. Date+time column format: YYYY-MM-DD hh:mm:ss; Concentrations column format: Float numbers (see Table 1).

TABLE 1: EXAMPLE OF INPUT DATA FOR THE PDA. FIRST COLUMN: DATE/TIME COLUMN, SECOND COLUMN: CONCENTRATIONS COLUMN. THE FIRST LINE WILL BE READ AS THE HEADER AND WILL BE OVERWRITTEN.

Date/Time	concentrations
YYYY-MM-DD hh:mm:ss	Float number
YYYY-MM-DD hh:mm:ss	Float number
YYYY-MM-DD hh:mm:ss	Float number

Running the Script

Figure 1 shows the various steps of the script in a diagram. More information can be found in Beck et al. (submitted to AMT.). A description of the single steps follows here. If more than one input parameter is asked, they should be separated by a space. As a rough estimate for some of the parameters, Table 2 summarizes the input parameters we used with the datasets to develop the PDA.

1. Enter the path to your datafile.
2. Enter the path to the directory to save plots and output data.
3. Enter the name of your datafile (incl. ending).
4. Do you wish to average your data? If No, type enter. If yes, type the averaging time in seconds for the gradient plot:
   - Enter a float or integer number, will be transformed to integer.
   - If you type in a number, the script will calculate the gradients of your original time resolution and average them afterwards to the input time resolution before continuing with the PDA. This is useful if your original data has a higher time resolution than actually needed for further analysis and allows the first step to take high fluctuations into account.

5. A new figure with the gradient vs concentration and a time series of your data was saved to your target directory. Please open it to make a decision for the next step.
    - A figure with the name "gradient_and_timeseries.png", followed by the date, hour and minute of the saving time will be saved in your target folder. Please check the figure in your target directory.
6. Step1: Continue with power law filter (a) or interquartile range filter (b)?
    - Type a or b.
    - Based on the saved figure, decide whether you want to continue with the power law filter or with the interquartile range filter. Figure 3 shows two examples. If your dataset looks like in a), we would use the power law filter, if it looks like in b) we would choose the interquartile range filter.
7. If you have selected option (a):
   Step 1A: Type two fix points (x1, y1) and (x2, y2) from the graph to find a separation line:
    - Enter X and Y coordinates of two points in the figure, where the line should lead through (rough estimate, the line can be adjusted later on). The script now determines the slope *m* and the intercept *a* of your line. The fixpoints should be entered as four numbers, separated with spaces (x1 y1 x2 y2). In the example from Figure 3, type: 100 10 10000 100

   Step 1A: Do you wish to enter a new slope and a new intercept for the separation line (y/n)?
    - Enter y for yes or n for no.
    - A figure with a provisional separation line, derived from the two fix points, was saved in the target directory (Fig. 3). The actual slope *m* and intercept *a* of this line are indicated in the figure.

   Step 1A. Actual intercept *a* and slope *m*: *a* = XX, *m* = YY. Type in the new values (*a m*):
    - Type in two float numbers, separated by a space.
    - The slope and the intercept can now be adjusted in order to move the line to where it separates the polluted from the clean mode of your data. This will save a new figure, and the process can be repeated until the user is satisfied (Fig. 4).

8. If you have selected option (b):
   Step 1B: Choose the IQR window size and the IQR factor:
    - Integer (minutes) and float (factor)
    - For 1 min particle number concentration data, we found 1440 minutes (24 h) and a factor of 1.7 to work. In that case, just type: 1440 1.7

9. Step 2: Choose the upper and the lower threshold
    - Enter two integer numbers, separated by a space. Example, type: 10000 60
    - Concentrations above the upper threshold will be flagged as polluted. Concentrations below the lower threshold will not be flagged as clean.
10. Step 3: Do you want to apply the neighboring points filter?
    - y for yes, n for no
11. Step 4: Do you want to apply the median filter?
    - Enter median time (in minutes) and median factor. Example, type: 30 1.5
12. Step 5: Enter the sparse filter window size and the threshold.

    - Size and threshold are a measure of the number of data points. For example, a window size of 30 data points and a threshold of 24 data points means that if there are more than 24 polluted data points within 30 points the sparse filter is applied. Example, type: 30 24

13. To check out the result, type in a start time and an end time to generate a time series plot. Example, type: 2015-01-01 2015-02-01
    - A figure will be saved to your target directory with 2 – 4 panels (depending on how many filter steps are activated), which shows you a time series in each panel of the original data (in red) and the "cleaned" data (in blue). The last panel always shows the result after the application of all the activated filter. These data will be saved in your target directory.
    - The final dataset will be saved in your target directory as a .csv file. It contains
        - Date/Time: Date and time of the data point
        - Concentration: Original concentration
        - Gradient: Calculated gradient for each data point
        - threshold_clean: Clean dataset after application of the gradient and threshold filters (Step 1 and 2)
        - threshold_flag (1=polluted): Flag after application of the gradient and threshold filters (Step 1 and 2). Polluted data points are flagged with 1, unaffected with 0.
        - neighbor_clean: Clean dataset after application of the median filter (Step 3)
        - median_clean: Clean dataset after application of the median filter (Step 4)
        - median_flag (1=polluted): Flag after application of the median filter (Step 4). Polluted data points are flagged with 1, unaffected with 0.
        - sparse_clean: Clean dataset after application of the sparse filter (Step 5). This is the final dataset
        - sparse_flag: Flag after application of the sparse data filter (Step 5). Polluted data points are flagged with 1, unaffected with 0. This is the final flag product.
    - Additionally, an excel spreadsheet will be saved with statistics of each filtering step. The first row in this sheet shows the number of data points after application of each individual filter. This is useful to track how many data points have been flagged as pollution.

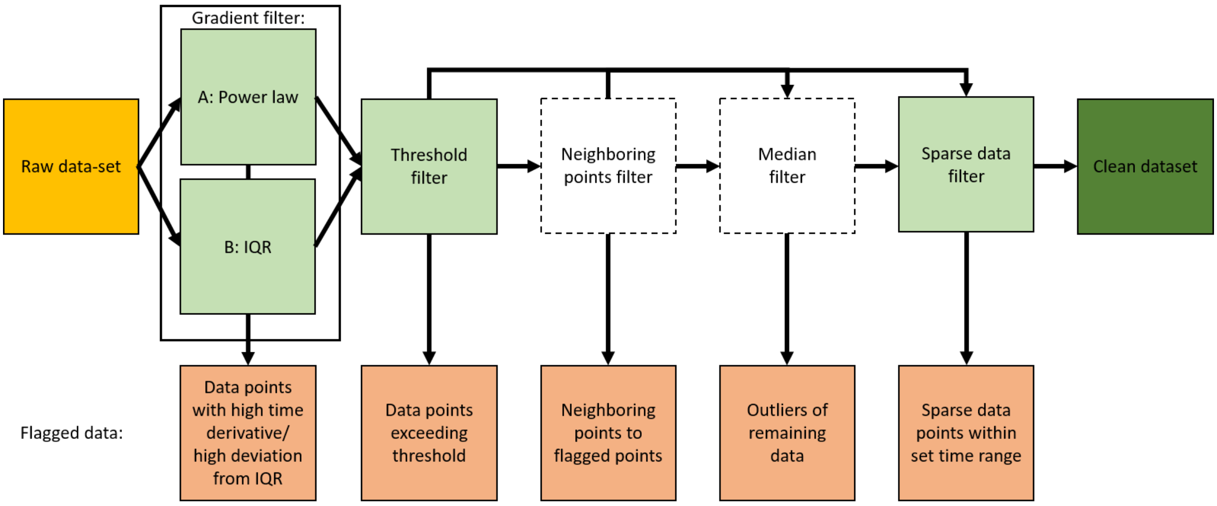
Gradient filter:
A: Power law
B: IQR
Raw data-set
Threshold filter
Neighboring points filter
Median filter
Sparse data filter
Clean dataset
Flagged data:
Data points with high time derivative/ high deviation from IQR
Data points exceeding threshold
Neighboring points to flagged points
Outliers of remaining data
Sparse data points within set time range

FIGURE 1: SCHEMATIC OF THE PDA SCRIPT FILTERING STEPS (BECK ET AL., SUBMITTED TO AMT)

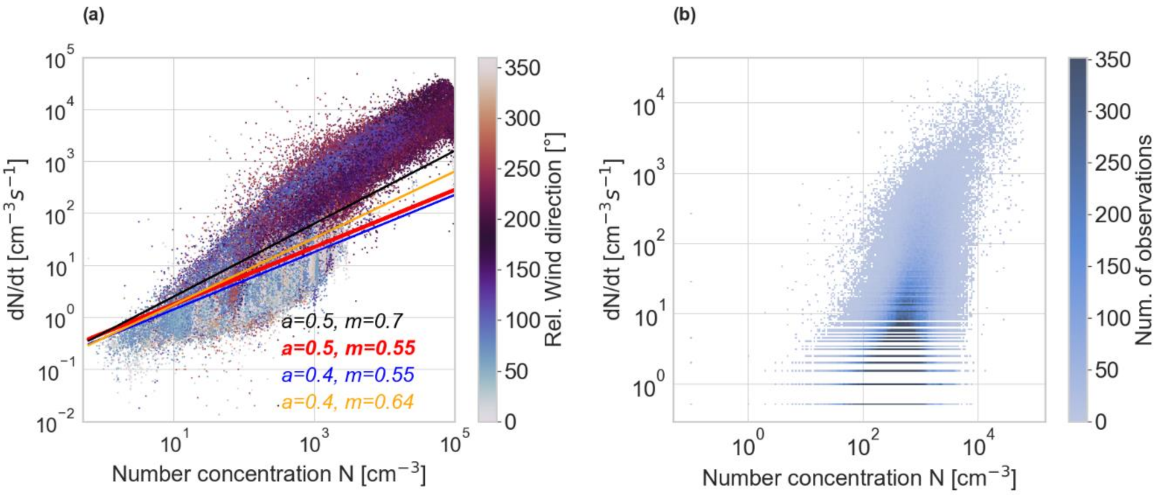

FIGURE 2: THE GRADIENT (Y-AXIS) VS. PARTICLE NUMBER CONCENTRATION (X-AXIS) PLOT OF TWO DIFFERENT EXAMPLE DATASETS. IN A, WE PLACED LINES WITH THE POWER LAW FILTER, IN B) WE WOULD CHOOSE THE IQR FILTER METHOD FOR THE GRADIENT FILTER (BECK ET AL., SUBMITTED TO AMT).

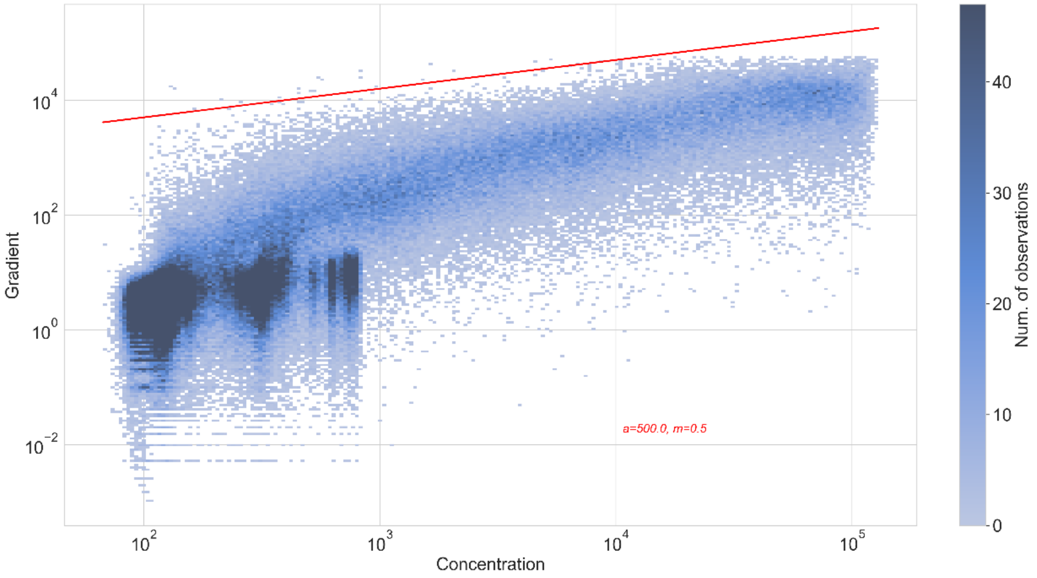

FIGURE 3: STEP 1A. THE GRADIENT VS CONCENTRATION PLOT WITH A SEPARATION LINE (IN RED) WITH A=500, M=0.5.

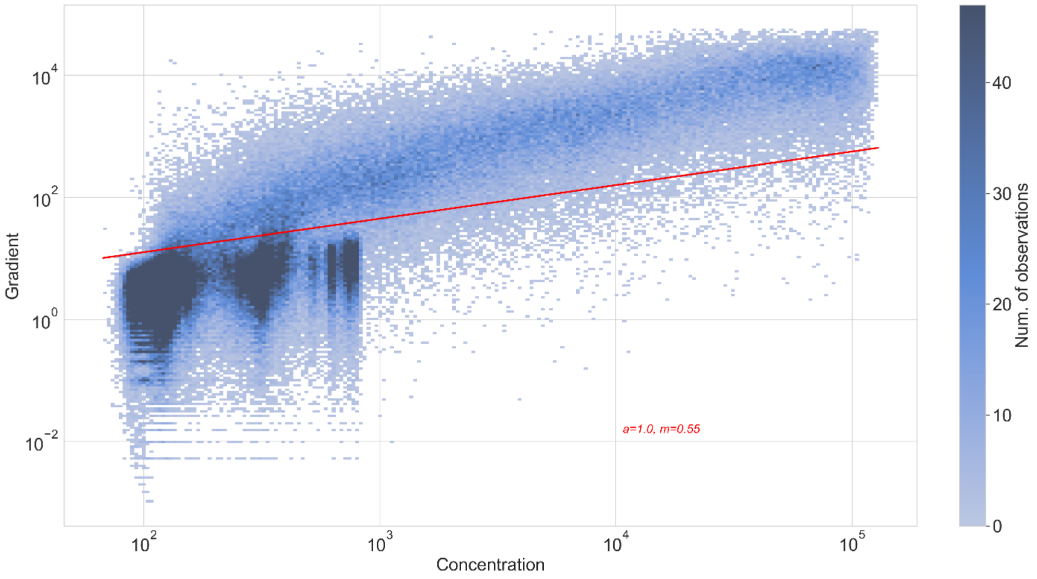

FIGURE 4: STEP 1A. AFTER ADJUSTING THE SEPARATION LINE.

TABLE 2: OVERVIEW OF THE PARAMETERS USED FOR SOME TESTED DATASETS. FOR MORE DETAILS, SEE BECK ET AL., SUBMITTED TO AMT.

Filter step	**Parameter**	**Particle number concentration MOSAiC**	**CO_2 MOSAiC dataset**	**Particle number concentration JFJ**
1A. Gradient filter (Power law)	a m	0.5 cm^{-3}s^{-1} 0.55 s^{-1}	- -	- -
1B. Gradient filter (IQR)	IQR factor IQR window size	- -	1.5 24 h	1.7 24 h
2. Threshold filter	Upper threshold Lower threshold	10^4 cm^{-3} 60 cm^{-3}	none none	10^4 cm^{-3} 60 cm^{-3}
3. Neighboring points filter	On/off	On	On	On
4. Median filter	Median time interval Median deviation factor	30 min 1.4	30 min 1.001	30 min 1.5
5. Sparse data filter (no. of data points)	Sparse window Sparse threshold	30 24	30 20	30 24

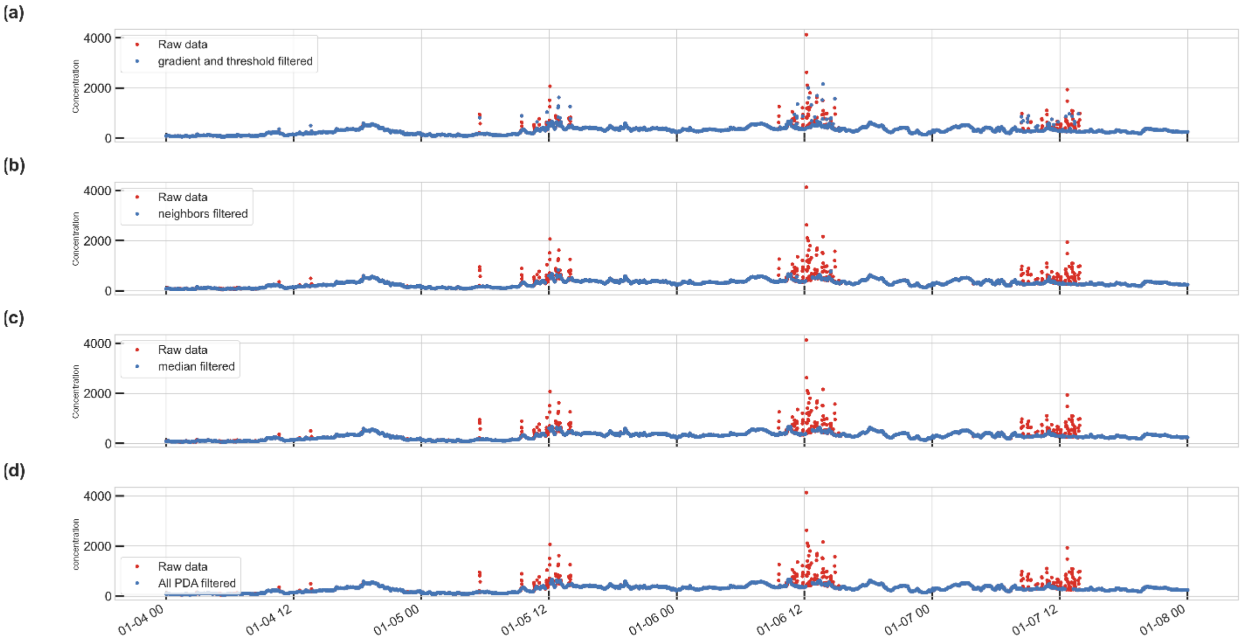

FIGURE 5: THE FINAL FIGURE WITH PANELS DEMONSTRATING THE DATA AFTER EACH FILTERING STEP. ORIGINAL DATA IS SHOWN IN RED, CLEANED DATA IN BLUE.

References

Beck, I., Angot, H., Dada, L., Baccarini, A., Quéléver, L. J., Jokinen, T., Laurila, T., Lampimaki, M., Bukowiecki, N., Boyer, M., Gong, X., Gysel-Beer, M., Petäjä, T., and Schmale, J.: Automated identification of local contamination in remote atmospheric composition time series, Atmos. Meas. Tech., Submitted in Dec. 2021.

Shupe, M. D., Rex, M., Blomquist, B., Persson, P. O. G., Schmale J., Uttal, T., Althausen, D., Angot, H., Archer, S., Bariteau, L., Beck, I., Bilberry, J., Bucci, S., Buck, C., Boyer, M., Brasseur, Z., Brooks, I. M., Calmer R., Cassano J., Castro V., Chu, D., Costa, D., Cox, C. J., Creamean, J., Crewell, S., Dahlke, S., Damm, E., de Boer, G., Deckelmann, H., Dethloff, K., Dütsch, M., Ebell, K., Ehrlich, A., Ellis, J., Engelmann, R., Fong, A. A., Frey, M. M., Gallagher, M. R., Ganzeveld, L.,Gradinger R., Graeser, J., Greenamyer, V., Griesche, H., Griffiths, S., Hamilton, J., Heinemann, G., Helmig, D., Herber A., Heuzé C. , Hofer, J., Houchens, T., Howard, D., Inoue J., Jacobi, H. W., , Jaiser, R., Jokinen, T., Jourdan, O., Jozef, G., King, W., Kirchgaessner, A., Klingebiel, M., Krassovski, M., Krumpen, T., Lampert, A., Landing, W., Laurila, T., Lawrence, D., Lonardi, M., Loose, B., Lüpkes, C., Maahn, M., Macke, A., Maslowski, W., Marsay, C., Maturilli, M., Mech, M., Morris, S., Moser, M., Nicolaus, M., Ortega, P., Osborn, J., Pätzold F., Perovich, D. K., Petäjä, T., Pilz, C., Pirazzini, R., Posman, K., Powers, H., Pratt, K. A., Preußer, A., Quéléver, L., Radenz, M., Rabe, B., Rinke, A., Sachs, T., Schulz, A., Siebert, H., Silva, T., Solomon, A., Sommerfeld, A., Spreen, G., Stephens, M., Stohl, A., Svensson, G., Uin, J., Viegas, J., Voigt, C., von der Gathen, P., Wehner, B., Welker, J. M., Wendisch, M., Werner, M., Xie, Z., and Yue, F.: Overview of the MOSAiC Expedition – Atmosphere, Elementa, 86, under review.