Supplement of Atmos. Meas. Tech., 17, 1363–1373, 2024
https://doi.org/10.5194/amt-17-1363-2024-supplement

Atmospheric Measurement Techniques
Open Access
EGU

Supplement of

A new software toolkit for optical apportionment of carbonaceous aerosol

Tommaso Isolabella et al.

Correspondence to: Federico Mazzei (federico.mazzei@ge.infn.it)

1 Toolkit implementation details (Python version)

The MWAA_MT toolkit is written entirely in Python 3.9 and makes use of standard libraries for scientific computing and data handling and visualization.

In particular, the data is read and manipulated using the Pandas library (Reback et al., 2022, v. 1.4.3), generic numeric tasks are handled with NumPy (Harris et al., 2020, v. 1.22.4). The power-law fits of analysis steps I and II are performed using the *optimize.curve_fit* function from the SciPy scientific computation library (Virtanen at al., 2020, v. 1.9.1), and linear regressions in steps I and IV are carried out with the *stats.linregress function* in SciPy.

Visualization of input data and of analysis results is done using the Matplotlib library (Hunter, 2007, v. 3.5.3).

2 Analysis configuration

The software can be customized to adapt the runtime analysis to the specific scenario. For example, users with no data for EC and OC concentrations can disable the mass apportionment stage of the analysis, while users can skip the preprocessing stage if independent (non-optical) measurements are not available. In addition, many parameters of each analysis stage can be tuned separately.

The configuration must be specified in a file written in the JSON format of *option_name:value*, usually named *config.json*. A template for such file, containing all the necessary options, can be found at https://github.com/tisolabella/mwaamt. The configuration file is divided into five sections, relating to the input and output file paths, the analysis steps to perform and their specific parameters.

The analysis necessitates one input data file, in .csv format, for which a template can be found at https://github.com/tisolabella/mwaamt/. The file has to contain the data in the format of one row per sample, with the first column being the sample ID, the subsequent columns containing the data for optical absorption at the wavelength indicated by each column header, and optionally the EC and OC concentrations and other available measurements that need to be considered in the analysis.

2.1 Pre-processing step: presets

For the optional analysis step I, a correlation with levoglucosan concentration measurements can be booked in order to optimize the values for α_{BC}, α_{FF} and α_{WB}.

- The levoglucosan correlation analysis optimizes α_{BC} by maximising the correlation between c_l, the levoglucosan concentration, and $b_{abs}^{BrC}(\lambda = \lambda_{min})$, α_{FF} and α_{WB} by maximising the correlation between c_l and $b_{abs}^{BC,WB}(\lambda = \lambda_{max})$.

2.2 Pre-processing step: α_{BC} swipe

As an additional check for the value of α_{BC} to be used in the subsequent analysis steps, an optional α_{BC} swipe analysis can be booked. It consists of varying α_{BC} in a wide range of values centred around 1.0. For each α_{BC} value, the fit in Eq. (1) is performed for all samples, then the average α_{BrC} and its standard deviation are computed, as shown in Fig. S1 for Propata and Milan. This analysis can give additional corroboration to a particular choice of a value for α_{BC} based on the

physical assumption that BrC in a particular location (especially rural areas) and time of year has a defined chemical composition and therefore its optical absorption (e.g., its α) should show little variation.

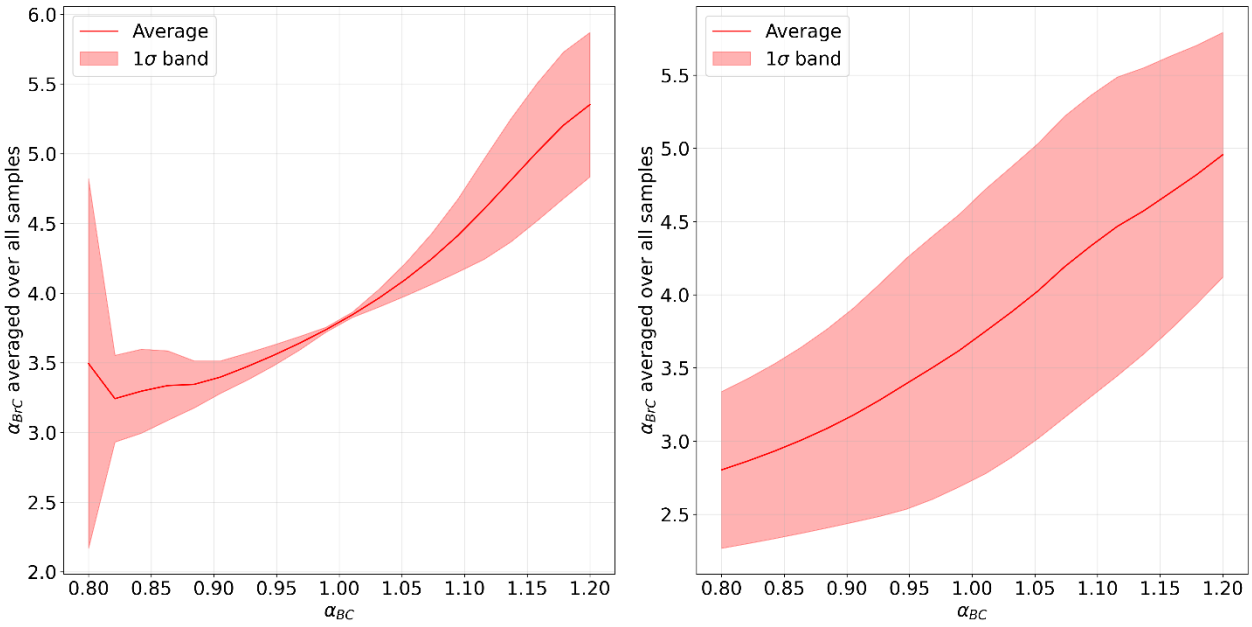

Figure S1: The α_{BC} swipe plots for the rural site (left) and the urban site (right). As can be seen by the width of the standard deviation band, this type of analysis corroborates the choice of $\alpha_{BC} = 1$ for the rural site, while proving inconclusive in the urban site where evidently the chemical composition of BrC is much more complex than in the rural site.

3 Toolkit output

The core of the analysis algorithm is represented by the optical apportionment procedure in steps II and III. Accordingly, the toolkit always provides one output .csv file for each of these runtime phases. For step II, the file `fitres.csv` is produced, which contains the fit parameters with their uncertainties for each sample; for step III, the file `appres.csv` is written, containing the apportioned optical absorption coefficient for each sample at all the wavelengths. Additionally, if step IV is enabled, a `mappres.csv` file is written with all the apportioned mass concentrations for each sample.

The toolkit always produces a `log.txt` file and a `data.pkl` file. In the log file, miscellaneous information is written, such as the input file, the working directory, the location of the written files and plots, as well as the results of the pre-processing step and the mass apportionment parameters k_1 and k_2. The data file is for internal use of the software and should be of no interest to the high-level user.

Finally, if the plots option is enabled in the configuration file, a number of subfolders containing plots for the various steps of the analysis will be created.

References

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Marchant, P. G., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T. E.: Array programming with NumPy. Nature 585, 357–362, https://doi.org/10.1038/s41586-020-2649-2, 2020.

Hunter, J. D.: "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, DOI: 10.1109/MCSE.2007.55, 2007.

Reback J., jbrockmendel, McKinney, W., Van den Bossche, J., Roeschke, M., Augspurger, T., Hawkins, S., Cloud, P., gfyoung, Sinhrks, Hoefler, P., Klein, A., Petersen, T., Tratner, J., She, C., Ayd, W., Naveh, S., Darbyshire, J. H. M., Shadrach, R., Li, T.: pandas-dev/pandas: Pandas 1.4.3 (v1.4.3). Zenodo [code]. https://doi.org/10.5281/zenodo.6702671, 2022.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272, https://doi.org/10.1038/s41592-019-0686-2, 2020