
SOFTWARE METAPAPER

An Open Source Software Suite for Multi-Dimensional Meteorological Data Computation and Visualisation

Y. Q. Wang

State Key Laboratory of Severe Weather and Key Laboratory of Atmospheric Chemistry of the China Meteorological Administration, Chinese Academy of Meteorological Sciences, Beijing, CN

yaqiang.wang@gmail.com

MeteoInfo Java software tools were developed for multi-dimensional meteorological data analysis and visualisation by integrating a Geographic Information System (GIS) and Scientific Computation Environment (SCE). Included are a Java class library for software developing, a GIS desktop application for spatial data operation and interactive multi-dimensional geoscientific data exploration, and a scientific computation and visualisation environment with Jython scripting. The popular geoscience data formats, such as NetCDF, HDF and GRIB, are supported based on a Unidata NetCDF Java library; also, its multi-dimensional array data model is used for scientific computation. In this paper, the software design framework and its implementation are presented. Furthermore, the software application capabilities are illustrated using several examples.

Keywords: Multi-dimensional data; Visualization; GIS; Scientific computation

(1) Overview

Introduction

Meteorological variables generally contain four dimensions of time and space (three dimensions), and more dimensions may be added for describing physical or chemical properties. Development of a scientific computation environment (SCE) with capabilities of multi-dimensional data computation, programming and visualisation is essential for meteorological and other scientific data analysis. The typical commercial one available is MATLAB (<https://www.mathworks.com/products/matlab.html>), developed by MathWorks Inc. to perform mathematical calculations, analyse and visualise data, and facilitate the writing of new software programs [1]. In the free and open source software (FOSS) field, the Python programming language with NumPy (<http://www.numpy.org>) and SciPy (<https://www.scipy.org>) extensions is a powerful environment for scientific computations with large datasets and complex computational programs [2], and its data visualisation capability was implemented by Matplotlib (<http://matplotlib.org>) and some other extensions. The PyAOS (Python for Atmosphere and Ocean Science) ecosystem of libraries built on top of NumPy, SciPy and Matplotlib is now quite extensive. Specified in meteorological fields, the Grid Analysis and Display System (GrADS) can perform multi-dimensional data computations through predefined dimension ranges, but multi-dimensional array operation functions are not included. The NCAR Command Language (NCL) provides a

powerful multi-dimensional array object with dimension type and value, which thus far has been able to process meteorological data more easily and powerfully.

Meteorological data are inherently variable, containing three spatial dimensions, and hence are suitable to be analysed using a Geographic Information System (GIS) with capabilities of powerful mapping, position-based information and analysis, topographic and land-use analysis, geostatistical analysis and other geoscientific methods and models [3, 4]. Besides commercial GIS software, such as ArcGIS, FOSS became a recognized counterpart to commercial solutions in the field of GIS and science [5, 6], such as GRASS GIS [7], QGIS [8], SAGA [9] and gvSIG [10]. GIS software focus on the data related to geographical position, but also time dimension integration has been an ongoing research theme [11]. Starting from 2010, the free software MeteoInfo was originally developed to promote the incorporation of GIS into meteorological fields through supporting basic GIS functions and widely used meteorological data formats, such as NetCDF and GRIB, using C# [4]. It has no SCE functions and has several deficiencies in terms of meteorological and GIS analysis: (1) a lack of 2-D and 3-D plot functions, except geo-mapping ability; (2) limited support of NetCDF, GRIB and HDF data formats; (3) a lack of editing and topology analysis GIS functions; (4) weak cross-platform ability.

Both SCE and GIS functions are important for meteorological research and applications, but they are

normally implemented in separated software tools. Integrating GIS and SCE functions in one framework can benefit users in terms of easily shifting between these two software environments and making the software more powerful for multiple purposes. To fill this gap, MeteoInfo was redeveloped with both an SCE and a GIS, using Java and Jython from 2014. Also, the new MeteoInfo software suite provides an interactive script development and running environment with GUI (Graphical User Interface), which is lacking in NCL and GrADS. Some other freely available software, vCDAT (<https://cdat.github.io/vcdat/docs/html/index.html>) and Panoply (<https://www.giss.nasa.gov/tools/panoply>), also provide GUI for exploration of meteorological data, but the operations are not in a powerful GIS environment. The MeteoInfo software suite is freely available via <http://www.meteothink.org>, and the source code is located at <https://github.com/meteoinfo> under an LGPL licence. Its design, capability and some application examples are presented in this paper.

Implementation/architecture

Software framework

Java was chosen as the programming language of MeteoInfo because of its powerful cross-platform capability and many existing open source libraries for

scientific data computation and other purposes. Jython (<http://www.jython.org>) is a Java implementation of Python that combines expressive power with clarity. It was used as a script language in MeteoInfo and to write the scientific computation and visualization packages through binding and extending the functions of the MeteoInfo Java library.

The MeteoInfo software framework is presented in **Figure 1**. There are three major components of: MeteoInfoLib, MeteoInfoMap and MeteoInfoLab. Among them, MeteoInfoLib is a Java class library with the functions of GIS, the reading of multiple data formats, multi-dimensional array computation, 2-D and 3-D plotting, and so on. The library was designed to implement the common functions used by MeteoInfoMap and MeteoInfoLab, which can also be used by other developers for multi-purpose software development.

MeteoInfoMap is a GIS desktop application based on MeteoInfoLib for end users. It has general GIS capabilities in terms of a layer-based spatial data view, geometry editing, spatial analysis, map composition and output. Also, an extra function of meteorological data exploration is included. MeteoInfoMap can be used to explore meteorological data interactively in a GIS environment.

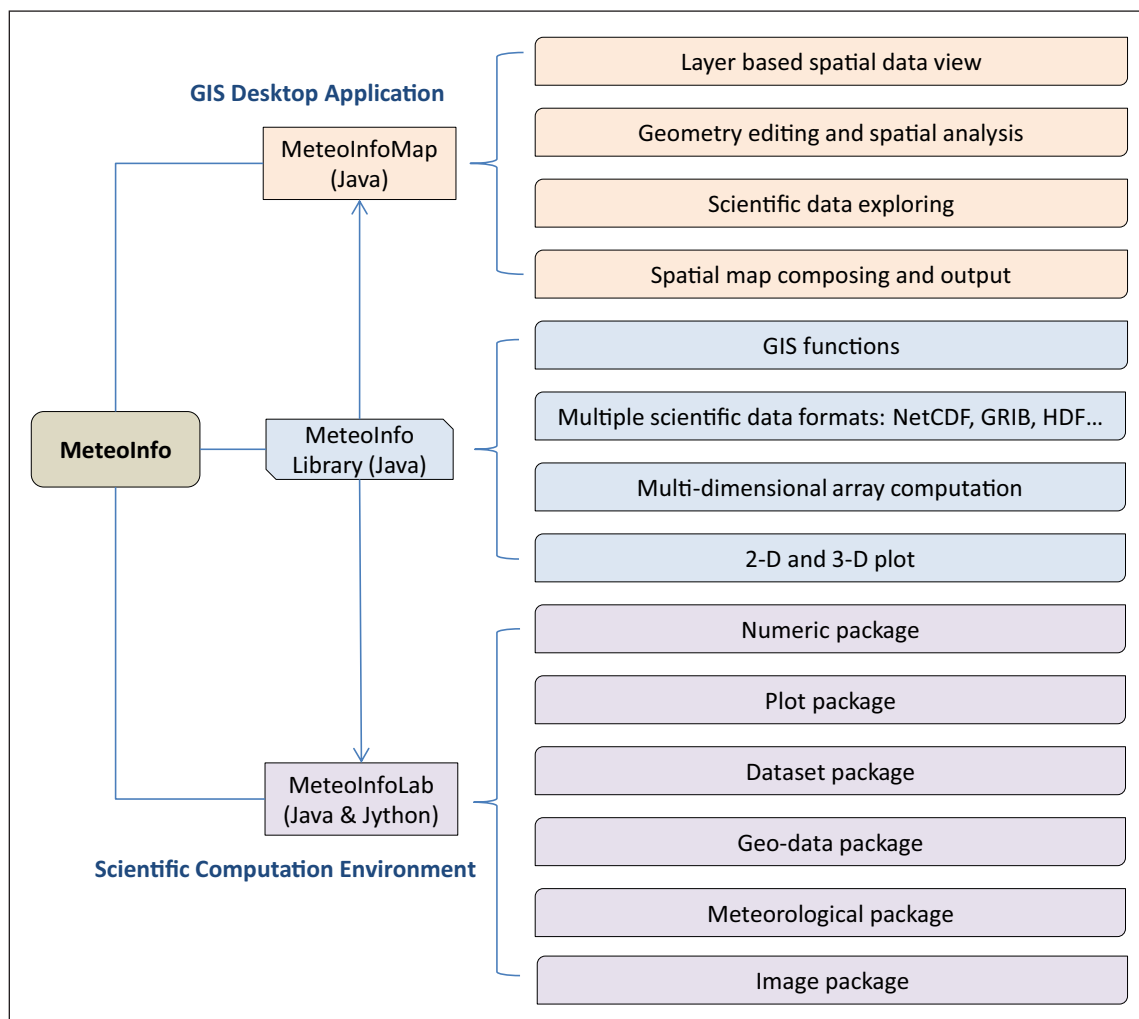


Figure 1: Framework of the MeteoInfo software suite and its main functions.

MeteoInfoLab is also for end users, as an SCE based on MeteoInfoLib. It includes an interactive Jython development environment application providing MATLAB-like features, and Jython extension packages for multi-dimensional array calculation, 2-D and 3-D plotting, scientific dataset input and output, geospatial data operation, meteorological data calculation and image processing.

The implementation of the key functions is described in the following sections.

General GIS functions

Some of the general GIS functions have already been presented in a paper on the .NET version of MeteoInfo [4], and thus only the different and extra functions are described in this paper. Geometry topology functions, such as buffer, convex hull and intersection, were implemented using JTS topology suite library (<https://www.locationtech.org/projects/technology.jts>), which provides algorithms in computational geometry. Some geometry editing and geoprocessing tools were provided in MeteoInfoMap to create vector layers as well as the ability to add and edit geometry objects interactively. Also, topology algorithms were used to support the geometry editing functions, such as geometry splitting, merging, etc. In MeteoInfoLab, a topology Jython package was developed to fulfil the geometry topology functions in the script environment.

The map layer object represents a single dataset with geometry collection or image and a basic display unit in a map. To provide more powerful geospatial background information, new web map layer is supported that loads web tile images from several web map providers, such as OpenStreetMap, BingMap, GoogleMap etc. The map view should be projected to Mercator projection to match

the web layer and other layers (**Figure 2**). Proj4J library (<https://trac.osgeo.org/proj4j/wiki>) code was used in MeteoInfoLib for projection function implementation.

2-D and 3-D plotting

General 2-D and 3-D plot functions are a basic requirement for multi-dimensional data visualisation. The basic chart types (line, scatter, bar, box, pie, etc.) and multiple coordinate systems (Cartesian, polar, geo-map and 3-D) are supported in the MeteoInfo Java library. The plot objects, with of point, polyline and polygon shape types, are created from the dataset, and then plotted in a chart panel with a variety of plot elements in terms of axis, text, legend, colorbar, and so on. Java code from SurfacePlotter (<https://github.com/ericaro/surfaceplotter>) was used for 3-D to 2-D coordinate projection calculation, so the 3-D objects can be plotted using Java2D. This is a lighter weight 3-D plot solution without OpenGL or other dependencies, and is therefore easy to deploy. With the provided basic 3-D plotting functions (scatter, surf, line, mesh), some properties of meteorological datasets can be viewed in a 3-D environment with a map content background, such as vertical and horizontal profiles or air mass trajectories and typhoon paths.

A gridded dataset is usually displayed as contour graphics through contour analysis, which is performed using the wContour Java library ported from the C# version [12]. The JLaTeXMath library (<https://github.com/opencollab/jlatexmath>) is used to display mathematical formulas written in LaTeX. The plotted charts can be exported as PNG, BMP or JPG image files; meanwhile, the vector formats of EPS and PDF vector formats are supported based on the FreeHEP VectorGraphics library (<http://java.freehep.org/vectorgraphics>).

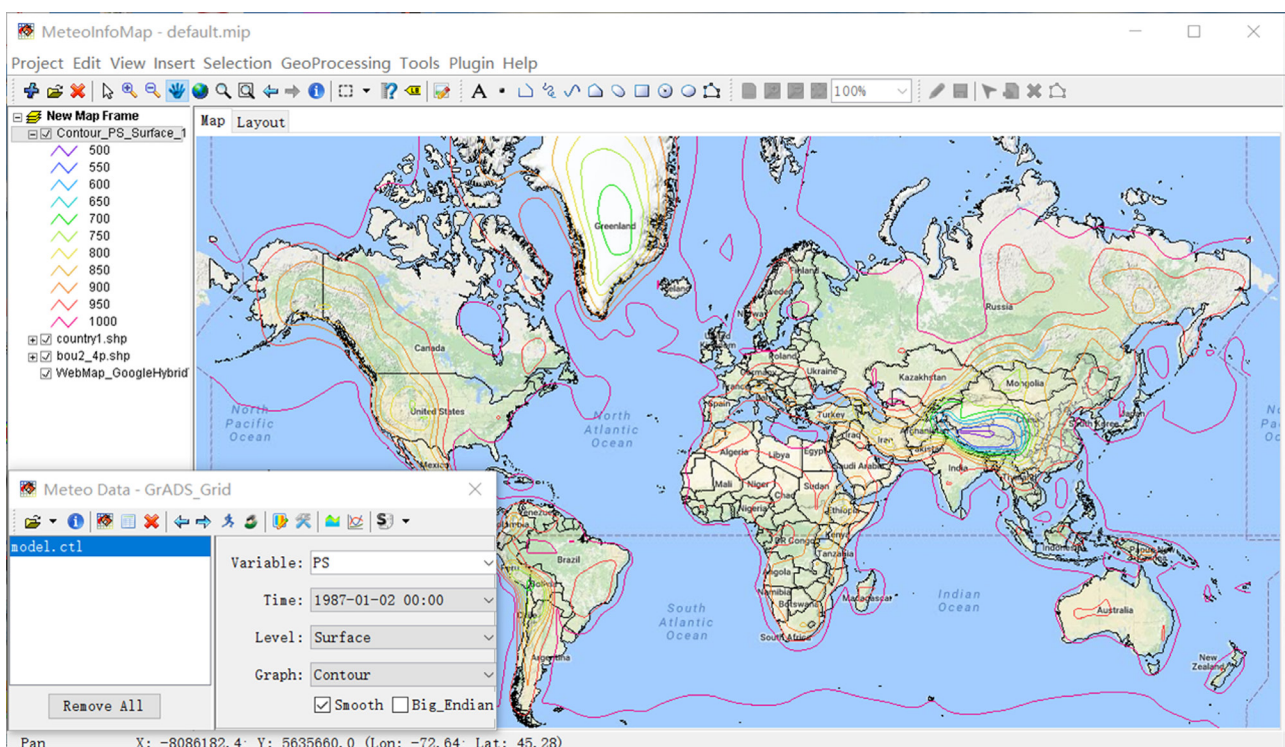


Figure 2: MeteoInfoMap GUI with web map and geoscience dataset exploration.

Geoscientific dataset

The widely used multi-dimensional geoscientific data format includes NetCDF [13], GRIB (<http://www.wmo.int/pages/prog/www/WDM/Guides/Guide-binary-2.html>) and HDF (<https://www.hdfgroup.org>). Unidata's Common Data Model (CDM) was developed as an abstract data model to create a common API for many types of scientific data [14]. The NetCDF Java library is an implementation of the CDM that can read these data formats and more, and is used in MeteoInfo for the implementation of scientific data input and output functions. Furthermore, MeteoInfoLib has the ability to read extra data formats, such as MICAPS, ARL and HYSPLIT model outputs [4].

Multi-dimensional array and its computation

An array is characterized by the type of elements it contains and by its shape. Geoscience data normally have three spatial dimensions or four dimensions with an extra time dimension. A multi-dimensional array data model has been successfully implemented in the NetCDF Java library with a maximum of seven dimensions and multiple data types. The subset data can be easily and effectively sliced using Array's section function or dimension index attribute. Based on this array data model, ArrayMath and ArrayUtil classes were developed in MeteoInfoLib for array computation. Universal mathematical functions are provided in ArrayMath class, while other array functions for array creation, broadcasting, interpolation, etc. are implemented in ArrayUtil class. The Apache Common Math library (<http://commons.apache.org/proper/commons-math>) is used for complex mathematical and statistical computations, such as linear algebra and curve fitting.

Scientific computation environment

MeteoInfoLab was developed as an SCE for Jython development (**Figure 3**). The GUI includes the docking components of console, editor, figure, file explorer and variable explorer, supported by the Docking Frames library (<http://www.docking-frames.org>). Jython's program editor was implemented by the RSyntaxTextArea library (<http://bobblylight.github.io/RSyntaxTextArea>) with Python syntax highlighting. The console part code from BeanShell (<http://www.beanshell.org>) was used for the Jython console in MeteoInfoLab. A draft code completion function was developed in the editor and console for users to write code more easily. The chart created from the code can be displayed in a figure panel that includes tools to zoom and pan the chart. File explorer is used to explore the files in a specific folder. The variables created during the execution of the code in editor or console can be explored in the variable explorer component. The functions of MeteoInfoLab can be extended by writing plugin applications using Java and/or Jython.

MeteoInfoLab Jython packages

Jython is a scripting language for using existing Java libraries with simple and easy-to-learn Python syntax. It can use scientific computation and plot functions directly from Java libraries, but there are some disadvantages of Java when used as a scientific computation language. For example, Java does not support operator overloading, which results in the inability to write mathematical formulas in a concise and user-friendly manner. Furthermore, Jython supports a variable number of arguments in functions, which is suitable for writing complex functions for scientific computation and

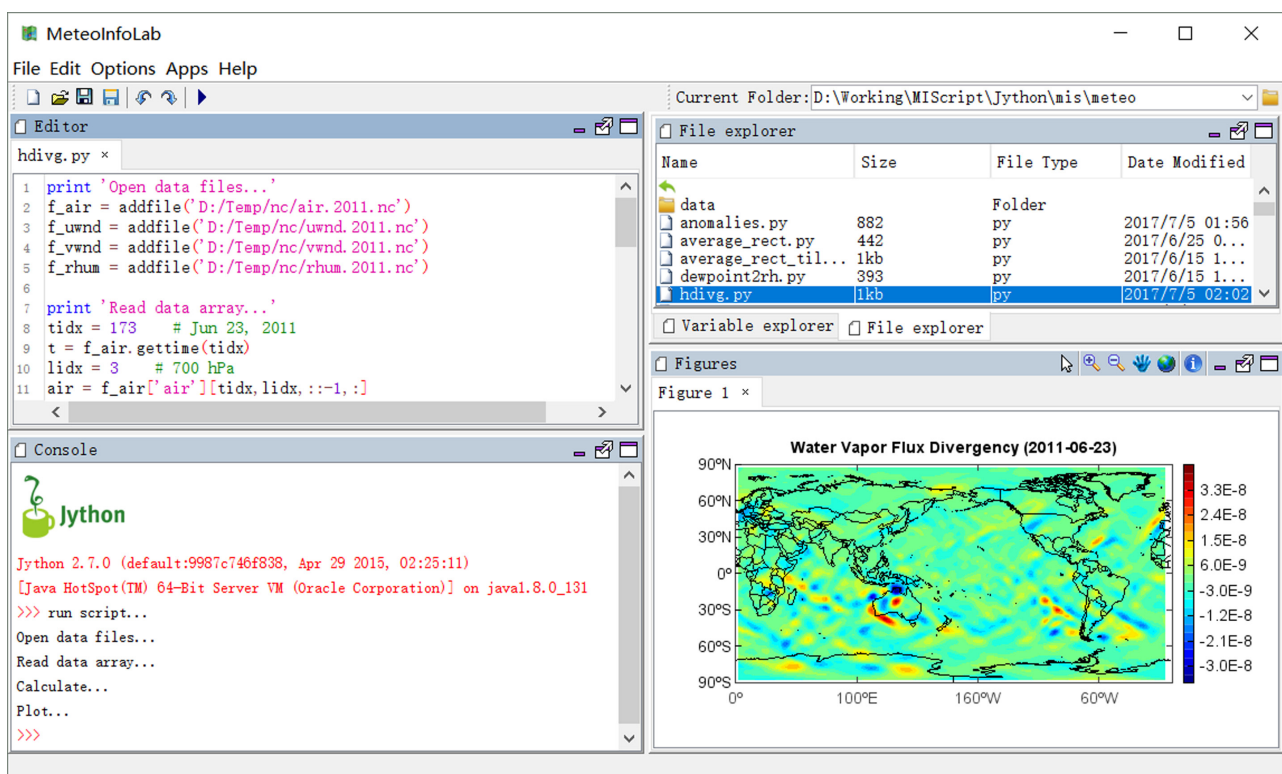


Figure 3: MeteoInfoLab GUI as a scientific computation environment.

visualisation. That is why MeteInfoLab Jython packages (**Figure 4**) were developed based on Java libraries, which run CPU-intensive tasks. The API mimics the semantics of Numpy and Matplotlib for easy of learning.

Numeric is the fundamental package for scientific computation in MeteInfoLab. The multi-dimensional array object can be created from the NDArry class, which contains an array object from the NetCDF Java library as described in section 2.5. The NDArry object has element-wise basic arithmetic operators and universal mathematical functions, and can be indexed, sliced and iterated over. Specified for the meteorological community, DimArray class was developed inherited from the NDArry class, a dims field containing dimension values and an optional geo-projection object, so the geolocation- and projection-based functions can be implemented. For example, the geoscientific array can be sliced by longitude/latitude ranges. PyTableData class was used to read ASCII table data files by column title, and temporal average functions are provided if the table has a time column. The array creation and computation functions are implemented in a minum module. The Numeric package also includes several preliminarily developed sub-packages with curve-fitting, data-interpolation,

linear-algebra, random-data-generation and statistical functions based on the Apache Common Math library.

All 2-D and 3-D plot functions are included in the plotlib package, which includes Axes, MapAxes, PolarAxes and Axes3D classes and a miplot module. A figure panel should first be created, and then one or more axes can be added in the figure, before finally plotting the data on the current axes by different plot functions. The Axes class represents a Cartesian coordinate set of axes, and is inherited by the PolarAxes, MapAxes and Axes3D classes to draw plot elements in polar, map and 3-D coordinate systems. One of the typical uses of the polar coordinate system in meteorological fields is for wind rose plots. Bases on the GIS functions of the MeteInfo Java library, the geospatial properties of meteorological parameters can be easily presented on a map with a certain projection. The multiple plot functions of plot, scatter, bar, boxplot, contour, etc. are implemented in the miplot module for user friendliness.

Another essential package for scientific data operation is "dataset". The addfile function in the midata module can create a DimDataFile object, including dimensions, variables and attribute information, from a NetCDF, GRIB or HDF scientific data file. Then, a DimVariable object

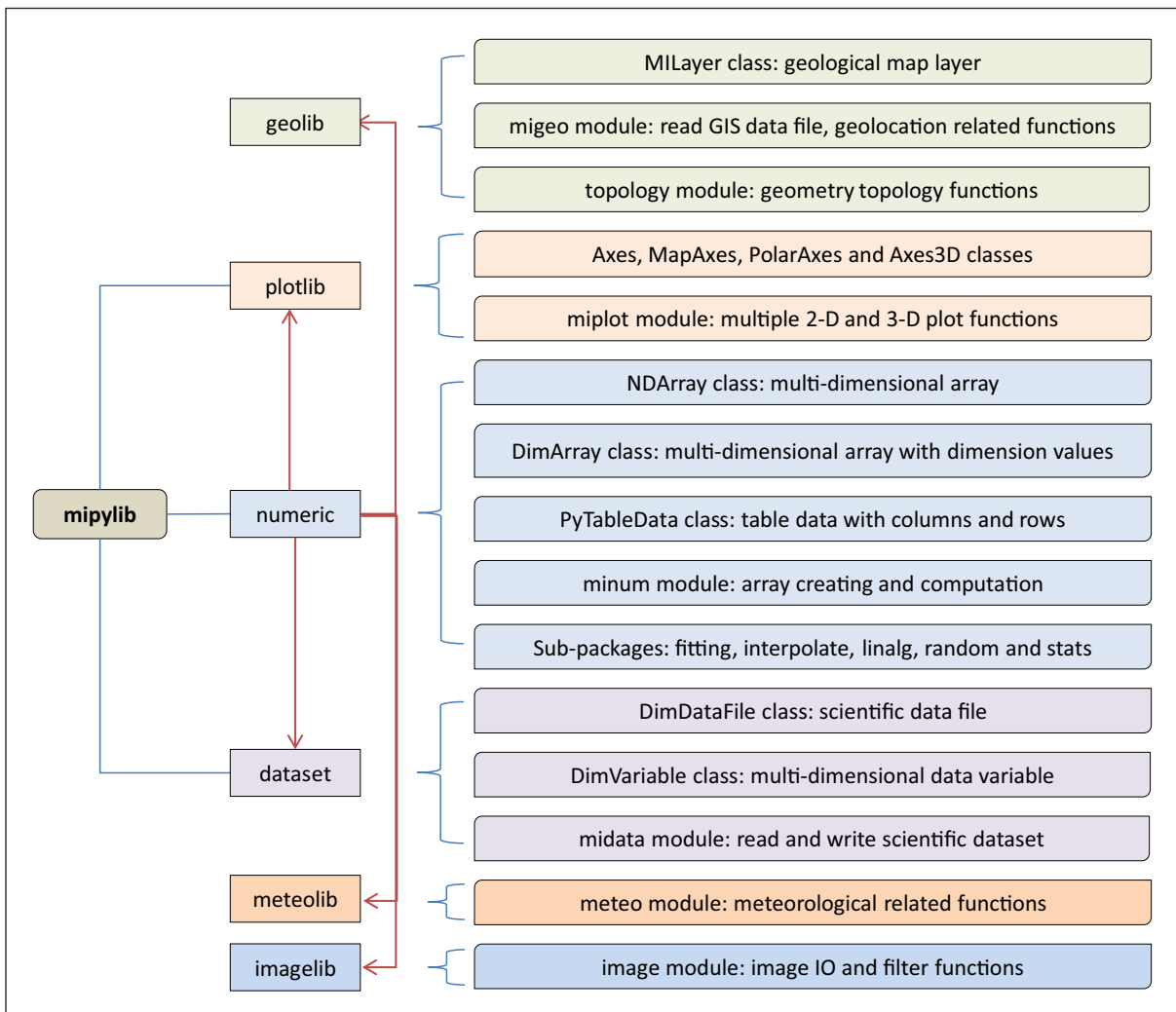


Figure 4: MeteInfoLab Jython packages, including the main modules and classes. The fundamental package is numeric, used by other packages.

can be read from the DimDataFile object by variable name. Finally, a multi-dimensional array (DimArray) is read from the DimVariable object through dimension slices.

The GIS related functions are implemented in the geolib package. A MLayer object can be created from two-dimensional geoscience data from the map plot functions in the miplot module, such as contourfm. The object contains the geometry collection and an associated attribute table, and behaves as a GIS layer. A GIS data file, such as a shape file, can be read as a MLayer object using the data reading functions of the migeo module, which also contains some geolocation-related functions, such as the maskout function to mask out the data with geopolygons. Geometry topology functions are implemented in the topology module based on the JTS library.

There are also two preliminary developed packages of meteolib and imagelib. Some meteorological-related functions are included in the meteo module of the meteolib package. In the imagelib package, the image module contains image reading, writing, and several image filtering functions based on Jerry's Java image processing library (<http://www.jhllabs.com/ip/index.html>).

Quality control

MeteoInfo has been functionally tested on the Windows, Mac OSX and Linux platforms to ensure consistent output. The computation and figure results of the main functions have been compared with other widely used meteorological data analysis software (i.e. GrADS and NCL). Some sample meteorological data files were included in the MeteoInfo application package, and sample script programs are provided on the MeteoInfo website for guiding software usage and quality checking. The website also provides many MeteoInfoLab API examples for software quality control. MeteoInfo has been widely used in the meteorological community, so users can greatly help in terms of software quality improvement.

(2) Availability

Operating system

As it is implemented in Java, MeteoInfo runs on any operating system that runs Java.

Programming language

Java 1.7 and Jython 2.7.1

Additional system requirements

None.

Dependencies

Using the application only requires the installation of a JRE. To engage with the source code, a number of maven dependencies are required. Using the provided pom.xml file in the repository and an IDE with maven functionality should automatically download the required packages.

List of contributors

Yaqiang Wang

Software location

Archive

Name: MeteoInfo

Persistent identifier: <https://doi.org/10.5281/zenodo.3251229>

License: GNU Lesser General Public License

Publisher: Yaqiang Wang

Version published: v1.8.5

Date published: Jun 21, 2019

Code repository

Name: GitHub

Identifier: <http://github.com/meteoinfo>

License: GNU Lesser General Public License

Date published: Jun 21, 2019

Language

English.

(3) Reuse potential

MeteoInfoMap and MeteoInfoLab were developed based on the MeteoInfoLib library which can also be used to develop other software with GIS and scientific computation and visualisation requirements. For example, the library was used for data plot functions in OutlierFlag, which is an open source software to flag the outlier values from scientific observation data [15]. It has also been used for developing the software of emissions inventory data processing, model output data verification, and so on.

MeteoInfoMap is a GIS desktop application with a user-friendly GUI and powerful geoscientific data exploration functions (**Figure 2**). Its main usages have been described in the paper on the .NET version of MeteoInfo [4], but the Java version is more powerful as a GIS application with geometry editing and topology functions, and web map layer support. It includes a framework for plugin development to extend its functions. TrajStat [16] was redeveloped as a MeteoInfoMap plugin and has been widely used as a tool to identify the source area of air pollution.

As an SCE emphasised in the geoscience field, MeteoInfoLab is useful for meteorological data analysis and visualisation. **Figure 3** includes an example of calculating the water vapor flux divergency from air temperature, relative humidity and u- and v-wind component data, and the result is plotted on map axes from a simple script program. It is a typical example of usage in the field of meteorology to study atmospheric features from a scientific dataset. Some example plots output from MeteoInfoLab script programs are given in **Figure 5**. Most popular 2-D plots are supported, such as line, bar, histogram, pie, box, error-bar, scatter, contour, image, etc. MeteoInfoLab could also be a powerful tool for satellite data processing, due to its ability to read HDF data formats and integrated GIS functions. For example, CAPLIPSO backscattering data were extracted from an HDF data file, and processed and plotted in the lower-left plot of **Figure 5**. Scientific data analysis functions are also illustrated in the middle-right plot of **Figure 5**, in which

the power-law fitting function was applied and plotted with LaTeX characters. Using matched polar axes and Cartesian axes, a special wind rose chart can be plotted using temporal observation data of wind direction, wind speed and PM_{10} concentration (top-centre plot in **Figure 5**). The chart is similar to a bivariate polar chart which is useful to identify and understand sources of air pollution [17]; moreover, the PM_{10} contour polygons were clipped by the convex hull polygon of the observation points and a further wind direction frequency line was added. The bottom-right plot of **Figure 5** shows the geometry topology functions of buffer and intersection. Meteoinfo was developed originally to be used in the field of meteorology, and so some specific meteorological plot functions were implemented. For example, the top-left plot of **Figure 5** was created by using the stationmodel plot function.

Two 3-D example plots are provided in **Figure 6**. Terrain height was plotted as 3-D surface and air mass back trajectories were plotted as 3-D lines, so the air mass transport directions and heights can be explored clearly. To study the vertical profile characteristics of the vertical velocity parameter along a line with a start point of (E80, N32) and end point of (E120, N40), oblique profile vertical velocity data were read and plotted on 3-D axes. Also, the terrain height along the line was plotted as masking.

Empirical orthogonal function (EOF) is one of the often-used analysis methods to study spatial patterns that vary with time from climate datasets. For example El Niño Southern Oscillation (ENSO) mode was revealed by an EOF analysis of winter sea surface temperature (SST) anomalies over the North Pacific [18]. Based on the eig and svd functions in the linalg sub-package for Eigen analysis and Singular Value Decomposition (SVD), the eof function was included in the meteo module of the meteoLib package to calculate EOF and PC (principal component) values by either the Eigen or SVD method. Missing values are removed automatically during EOF computation and placed back into the output data. A spatiotemporal transform option is provided in the Eigen analysis method to speed up the computation if the space location number is much more than the time stamps. To validate the EOF function, an example from a published Python library of eofs [19] was used with a dataset of November March averages of SST anomalies in the central and northern Pacific from 1963 to 2012 (**Figure 7**). The spatial pattern of the leading EOF is the canonical El Niño pattern, and the associated time series shows large peaks and troughs for well-known El Niño and La Niña events.

The scripts and figures of these examples and more can be found in the 'example section' of Meteoinfo website. MeteoinfoLab is also a highly all-purpose software

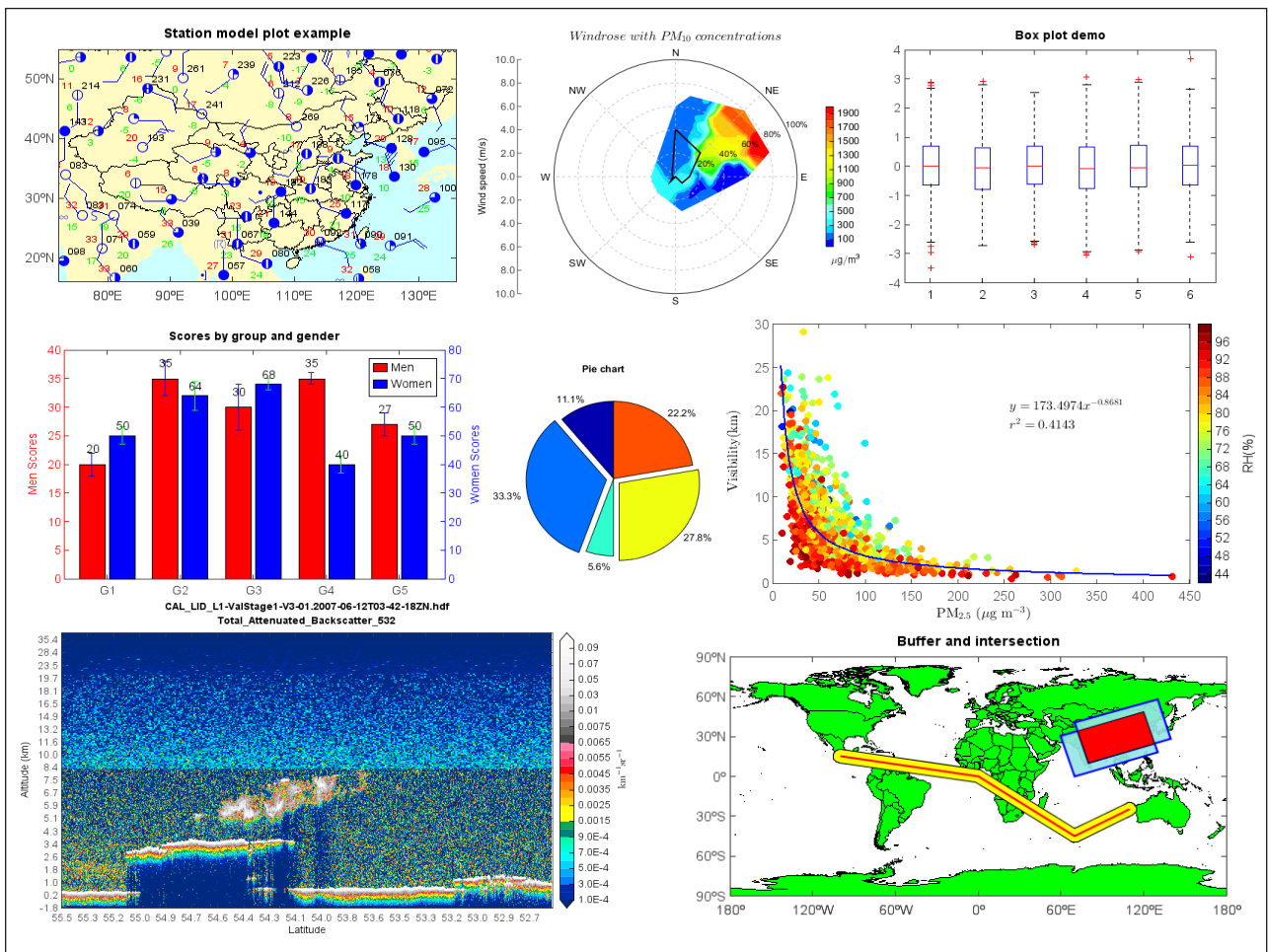


Figure 5: Example output plots from MeteoinfoLab script.

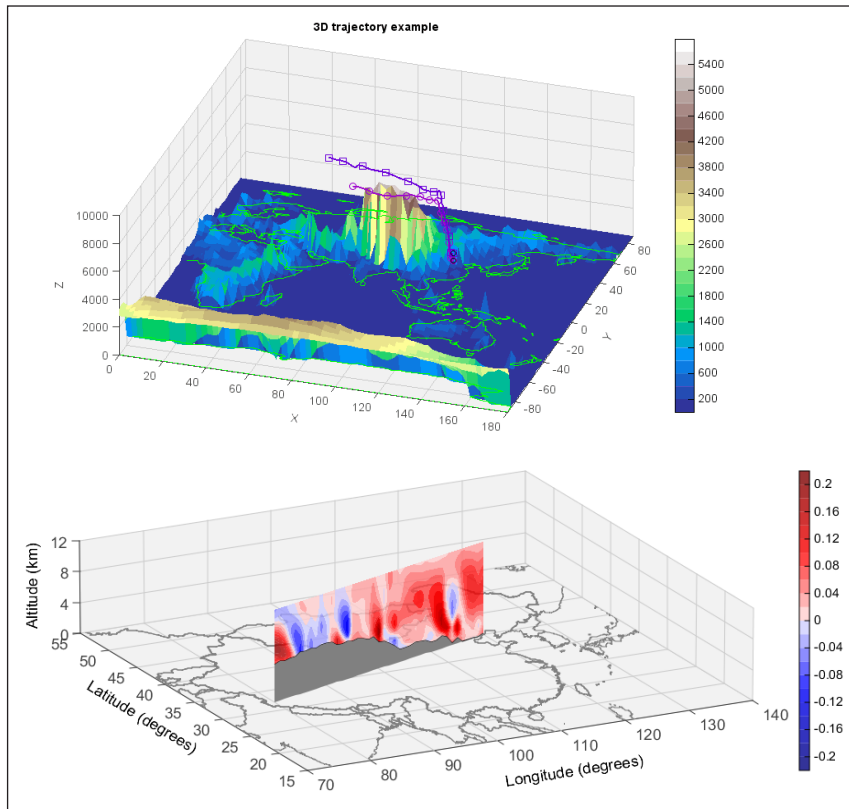


Figure 6: Air mass trajectory and vertical velocity 3-D example plots.

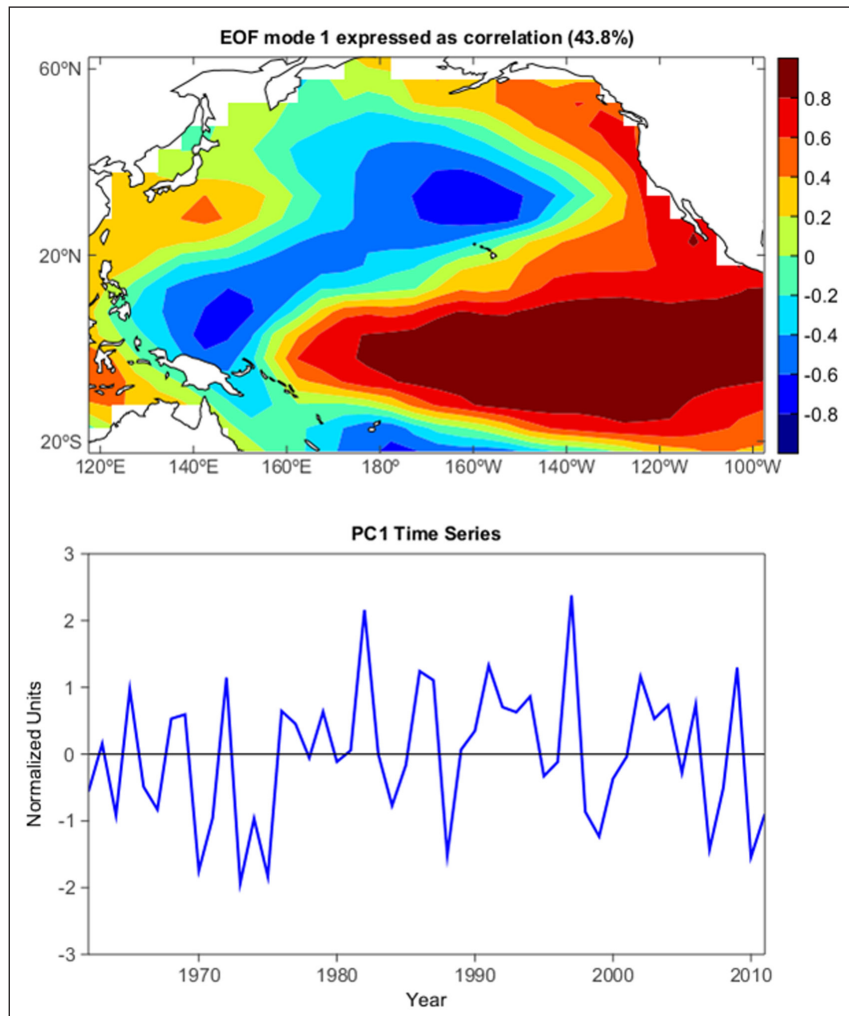


Figure 7: Leading EOF of SST in the central and northern Pacific during winter.

package, and thus has the potential to be used in many scientific fields. MeteoinfoLab provides a strategy to write add-on applications and toolboxes to extend its functions.

Acknowledgements

This research was supported by grants from the National Key Foundation Study Developing Programs (2016YFC0203306), the Atmospheric Pollution Control of the Prime Minister Fund (DQGG0104), and CAMS projects (2017Z001, 2018KJ002, and 2019Z014). Thanks to the authors and agencies of the open source software used by Meteoinfo.

Competing Interests

The author has no competing interests to declare.

References

1. **Trauth, M H** and **Sillmann, E** 2013 *MATLAB® and Design Recipes for Earth Sciences*. Berlin, Heidelberg: Springer-Verlag. DOI: <https://doi.org/10.1007/978-3-642-32544-1>
2. **Oliphant, T E** 2007 Python for scientific computing. *Computing in Science & Engineering*, 9: 10–20. DOI: <https://doi.org/10.1109/MCSE.2007.58>
3. **Chapman, L** and **Thornes, J E** 2003 The use of geographical information systems in climatology and meteorology. *Progress in Physical Geography*, 27(3): 313–330. DOI: <https://doi.org/10.1191/030913303767888464>
4. **Wang, Y Q** 2014 Meteoinfo: GIS software for meteorological data visualization and analysis. *Meteorological Applications*, 21(2): 360–368. DOI: <https://doi.org/10.1002/met.1345>
5. **Steiniger, S** and **Hunter, A J S** 2013 The 2012 free and open source GIS software map – A guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*, 39: 136–150. DOI: <https://doi.org/10.1016/j.compenurbsys.2012.10.003>
6. **Steiniger, S** and **Bocher, E** 2009 An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science*, 23(10): 1345–1370. DOI: <https://doi.org/10.1080/13658810802634956>
7. **Neteler, M**, et al. 2012 GRASS GIS: A multi-purpose open source GIS. *Environmental Modelling & Software*, 31: 124–130. DOI: <https://doi.org/10.1016/j.envsoft.2011.11.014>
8. **Hugentobler, M** 2008 Quantum GIS. In: *Encyclopedia of GIS*, Shekhar, S and Xiong, H (eds.), 935–939. New York: Springer. DOI: https://doi.org/10.1007/978-0-387-35973-1_1064
9. **Conrad, O**, et al. 2015 System for Automated Geoscientific Analyses (SAGA) v. 2.1.4. *Geoscientific Model Development*, 8: 1991–2007. DOI: <https://doi.org/10.5194/gmd-8-1991-2015>
10. **Anguix, A** and **Díaz, L** 2008 gvSIG: A GIS desktop solution for an open SDI. *Journal of Geography and Regional Planning*, 1(3): 41–48.
11. **Gebbert, S** and **Pebesma, E** 2014 A temporal GIS for field based environmental modeling. *Environmental Modelling & Software*, 53: 1–12. DOI: <https://doi.org/10.1016/j.envsoft.2013.11.001>
12. **Wang, Y Q** 2012 wContour: A .NET class library of contour-related algorithms. *Computers & Geosciences*, 48: 330–333. DOI: <https://doi.org/10.1016/j.cageo.2011.12.003>
13. **Rew, R** and **Davis, G** 1990 NetCDF: An interface for scientific data access. *IEEE Computer Graphics & Applications*, 10(4): 76–82. DOI: <https://doi.org/10.1109/38.56302>
14. **Nativi, S**, et al. 2008 Unidata's Common Data Model mapping to the ISO 19123 Data Model. *Earth Science Informatics*, 1(2): 59–78. DOI: <https://doi.org/10.1007/s12145-008-0011-6>
15. **Huang, S**, et al. 2016 OutlierFlag: A Tool for Scientific Data Quality Control by Outlier Data Flagging. *Journal of Open Research Software*, 4: e20. DOI: <https://doi.org/10.5334/jors.90>
16. **Wang, Y Q**, **Zhang, X Y** and **Draxler, R** 2009 TrajStat: GIS-based software that uses various trajectory statistical analysis methods to identify potential sources from long-term air pollution measurement data. *Environmental Modelling & Software*, 24(8): 938–939. DOI: <https://doi.org/10.1016/j.envsoft.2009.01.004>
17. **Carlaw, D C** and **Ropkins, K** 2012 openair – An R package for air quality data analysis. *Environmental Modelling & Software*, 27–28: 52–61. DOI: <https://doi.org/10.1016/j.envsoft.2011.09.008>
18. **Deser, C** and **Blackmon, M L** 1995 On the relationship between tropical and north Pacific sea surface temperature variations. *Journal of Climate*, 8: 1677–1680. DOI: [https://doi.org/10.1175/1520-0442\(1995\)008<1677:OTRBTA>2.0.CO;2](https://doi.org/10.1175/1520-0442(1995)008<1677:OTRBTA>2.0.CO;2)
19. **Dawson, A** 2016 eofs: A Library for EOF Analysis of Meteorological, Oceanographic, and Climate Data. *Journal of Open Research Software*, 4: e14. DOI: <https://doi.org/10.5334/jors.122>

How to cite this article: Wang, Y Q 2019 An Open Source Software Suite for Multi-Dimensional Meteorological Data Computation and Visualisation. *Journal of Open Research Software*, 7: 21. DOI: <https://doi.org/10.5334/jors.267>

Submitted: 06 March 2019 **Accepted:** 28 June 2019 **Published:** 11 July 2019

Copyright: © 2019 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.