
Pyleoclim Documentation

Release 0.1.4

Deborah Khider, Julien Emile-Geay

February 28, 2017

CONTENTS

1	Pyleoclim	3
1.1	What is it?	3
1.2	Version Information	3
1.3	Installation	4
1.4	Quickstart guide	4
1.5	Requirements	4
1.6	Further information	4
1.7	Contact	4
1.8	License	5
1.9	Disclaimer	5
2	Main Functions	7
2.1	Getting started	7
2.2	Mapping	8
2.3	Plotting	9
2.3.1	Summary Plots	11
2.4	Statistics	12
3	Basic Module	15
4	Mapping Module	17
5	Timeseries Plots Module	19
6	Summary Plots Module	21
7	Manipulating LiPD files	25
7.1	Creating directories and saving figures	25
7.2	Manipulating LiPD files	25
7.3	Manipulating Variables in a LiPD file	26
7.4	Manipulating timeseries objects	26
7.5	Handling mapping to LinkedEarth Ontology	27
8	Indices and tables	29
	Index	31

Contents:

PYLEOCLIM

1.1 What is it?

Pyleoclim is a Python package primarily geared towards the analysis and visualization of paleoclimate data. Such data often come in the form of timeseries with missing values and age uncertainties, and the package includes several low-level methods to deal with these issues, as well as high-level methods that re-use those to perform scientific workflows.

The package assumes that the data are stored in the Linked Paleo Data ([LiPD](#)) format and makes extensive use of the [LiPD utilities](#). The package is aware of age ensembles stored via LiPD and uses them for time-uncertain analyses very much like [GeoChronR](#).

Current Capabilities:

- binning
- interpolation
- plotting maps, timeseries, and basic age model information

Future capabilities:

- paleo-aware correlation analysis (isopersistent, isospectral, and classical t-test)
- paleo-aware singular spectrum analysis (AR(1) null eigenvalue identification, missing data)
- spectral analysis (Multi-Taper Method, Lomb-Scargle)
- weighted wavelet Z transform (WWZ)
- cross-wavelet analysis
- index reconstruction
- climate reconstruction
- ensemble methods for most of the above

1.2 Version Information

0.1.4: Rename functions using camel case convention and consistency with LiPD utilities version 0.1.8.5

0.1.3: Compatible with LiPD utilities version 0.1.8.5

Function `openLiPD()` renamed `openLiPDs()`

0.1.2: Compatible with LiPD utilities version 0.1.8.3

Uses Basemap instead of cartopy

0.1.1: Freezes the package prior to version 0.1.8.2 of LiPD utilities

0.1.0: First release

1.3 Installation

Python v3.5+ is required Pyleoclim is published through Pypi and easily installed via pip:

```
pip install pyleoclim
```

1.4 Quickstart guide

1. Open your command line application (Terminal or Command Prompt)
2. Install with command:

```
pip install pyleoclim
```
3. Wait for installation to complete, then:
 1. Import the package into your favorite Python environment (we recommend the use of Spyder, which comes standard with the Anaconda build)
 2. Use Jupyter Notebook to go through the tutorial contained in the [PyleoclimQuickstart.ipynb](#)

1.5 Requirements

- LiPD v0.1.8.5
- pandas v0.19+
- numpy v1.12+
- matplotlib v2.0+
- basemap v1.0.7+

The installer will automatically check for the needed updates.

1.6 Further information

GitHub: https://github.com/LinkedEarth/Pyleoclim_util

LinkedEarth: <http://linked.earth>

Python and Anaconda: <http://conda.pydata.org/docs/test-drive.html>

Jupyter Notebook: <http://jupyter.org/>

1.7 Contact

Please report issues to linkedearth@gmail.com

1.8 License

The project is licensed under the [GNU Public License](#) .

1.9 Disclaimer

This material is based upon work supported by the U.S. National Science Foundation under Grant Number ICER-1541029. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the investigators and do not necessarily reflect the views of the National Science Foundation.

MAIN FUNCTIONS

2.1 Getting started

Pyleoclim relies heavily on the concept of timeseries objects introduced in [LiPD](#) and implemented in the [LiPD utilities](#). Briefly, timeseries objects are dictionaries containing the ChronData values and PaleoData values as well as the meta-data associated with the record. If one record has three ProxyObservations (e.g., Mg/Ca, d18O, d13C) then it will have three timeseries objects, one for each of the observations.

The LiPD utilities function `lipd.extractTs()` returns a list of dictionaries for the selected LiPD files, which need to be passed to Pyleoclim along with the path to the directory containing the LiPD files.

This is done through the function `pyleoclim.openLiPDs`:

```
pyleoclim.openLiPDs (path='', ts_list='')
```

Load and extract timeseries objects from LiPD files.

Allows to load and extract timeseries objects into the workspace for use with Pyleoclim. This can be done by the user previously, using the LiPD utilities and passed into the function's arguments. If no timeseries objects are found by other functions, this function will be triggered automatically without arguments.

Parameters

- **path** (*string*) – the path to the LiPD file. If not specified, will trigger the LiPD utilities GUI.
- **ts_list** (*list*) – the list of available timeseries objects obtained from `lipd.extractTs()`.

Warning: if specifying a list, path should also be specified.

Examples

```
>>> pyleoclim.openLiPDs(path = "/Users/deborahkhider/Documents/LiPD")
Found: 12 LiPD file(s)
processing: Crystal.McCabe-Glynn.2013.lpd
processing: MD01-2412.Harada.2006.lpd
processing: MD98-2170.Stott.2004.lpd
processing: MD982176.Stott.2004.lpd
processing: O2kLR-EmeraldBasin.Sachs.2007.lpd
processing: Ocean2kHR-AtlanticBahamasTOTORosenheim2005.lpd
processing: Ocean2kHR-AtlanticCapeVerdeMoses2006.lpd
processing: Ocean2kHR-AtlanticMontegoBayHaaseSchramm2003.lpd
processing: Ocean2kHR-AtlanticPrincipeSwart1998.lpd
processing: Ocean2kHR-PacificClippertonClipp2bWu2014.lpd
```

```
processing: Ocean2kHR-PacificNauruGuilderson1999.lpd
processing: ODP1098B.lpd
extracting: ODP1098B.lpd
extracting: MD98-2170.Stott.2004.lpd
extracting: Ocean2kHR-PacificClippertonClipp2bWu2014.lpd
extracting: Ocean2kHR-AtlanticBahamasTOTORosenheim2005.lpd
extracting: Ocean2kHR-AtlanticPrincipeSwart1998.lpd
extracting: Ocean2kHR-AtlanticMontegoBayHaaseSchramm2003.lpd
extracting: MD982176.Stott.2004.lpd
extracting: Ocean2kHR-PacificNauruGuilderson1999.lpd
extracting: O2kLR-EmeraldBasin.Sachs.2007.lpd
extracting: Crystal.McCabe-Glynn.2013.lpd
extracting: Ocean2kHR-AtlanticCapeVerdeMoses2006.lpd
extracting: MD01-2412.Harada.2006.lpd
Finished time series: 31 objects
Process Complete
```

2.2 Mapping

`pyleoclim.mapAll (markersize=50, saveFig=False, dir='', format='eps')`

Map all the available records loaded into the workspace by archiveType.

Map of all the records into the workspace by archiveType. Uses the default color palette. Enter `pyleoclim.plot_default` for detail.

Parameters

- **markersize** (*int*) – The size of the markers. Default is 50
- **saveFig** (*bool*) – Default is to not save the figure
- **dir** (*str*) – The absolute path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (`lipd.path`).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure

Examples

```
>>> fig = pyleoclim.mapAll()
```

```
pyleoclim.mapLipd(name='', countries=True, counties=False, rivers=False, states=False, back-
ground='shadedrelief', scale=0.5, markersize=50, marker='default', save-
Fig=False, dir='', format='eps')
```

Create a Map for a single record

Orthographic projection map of a single record.

Parameters

- **name** (*str*) – the name of the LiPD file. **WITH THE .LPD EXTENSION!**. If not provided, will prompt the user for one
- **countries** (*bool*) – Draws the country borders. Default is on (True).
- **counties** (*bool*) – Draws the USA counties. Default is off (False).

- **states** (*bool*) – Draws the American and Australian states borders. Default is off (False)
- **background** (*str*) – Plots one of the following images on the map: bluemarble, etopo, shadedrelief, or none (filled continents). Default is shadedrelief
- **scale** (*float*) – useful to downgrade the original image resolution to speed up the process. Default is 0.5.
- **markersize** (*int*) – default is 100
- **marker** (*str*) – a string (or list) containing the color and shape of the marker. Default is by archiveType. Type `pyleo.plot_default` to see the default palette.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (`lipd.path`).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure

Examples

```
>>> fig = pyleoclim.mapLipd(markersize=100)
```

2.3 Plotting

```
pyleoclim.plotTs(timeseries='', x_axis='', markersize=50, marker='default', saveFig=False, dir='',
                  format='eps')
```

Plot a single time series.

Parameters

- **timeseries** (*A*) – By default, will prompt the user for one.
- **x_axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **markersize** (*int*) – default is 50.
- **marker** (*str*) – a string (or list) containing the color and shape of the marker. Default is by archiveType. Type `pyleo.plot_default` to see the default palette.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (`lipd.path`).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure.

Examples

```
>>> fig = pyleoclim.plotTs(marker = 'rs')
```

```
pyleoclim.histTs(timeseries='', bins=None, hist=True, kde=True, rug=False, fit=None,
                 hist_kws={'label': 'hist'}, kde_kws={'label': 'kde'}, rug_kws={'label': 'rug'},
                 fit_kws={'label': 'fit'}, color=None, vertical=False, norm_hist=True, legend=True,
                 saveFig=False, format='eps', dir='')
```

Plot a univariate distribution of the PaleoData values

This function is based on the seaborn displot function, which is itself a combination of the matplotlib hist function with the seaborn kdeplot() and rugplot() functions. It can also fit scipy.stats distributions and plot the estimated PDF over the data.

Parameters

- **timeseries** – A timeseries. By default, will prompt the user for one.
- **bins** (*int*) – Specification of hist bins following matplotlib(hist), or None to use Freedman-Diaconis rule
- **hist** (*bool*) – Whether to plot a (normed) histogram
- **kde** (*bool*) – Whether to plot a gaussian kernel density estimate
- **rug** (*bool*) – Whether to draw a rugplot on the support axis
- **fit** – Random variable object. An object with fit method, returning a tuple that can be passed to a pdf method of positional arguments following a grid of values to evaluate the pdf on.
- **kde, rug, fit}_kws** (*{hist,}*) – Dictionaries. Keyword arguments for underlying plotting functions. If modifying the dictionary, make sure the labels “hist”, “kde”, “rug” and “fit” are still passed.
- **color** (*str*) – matplotlib color. Color to plot everything but the fitted curve in.
- **vertical** (*bool*) – if True, observed values are on y-axis.
- **norm_hist** (*bool*) – If True (default), the histogram height shows a density rather than a count. This is implied if a KDE or fitted density is plotted
- **legend** (*bool*) – If true, plots a default legend
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns fig - The figure

Examples

```
>>> fig = pyleoclim.histTs(vertical = True)
```

2.3.1 Summary Plots

Summary plots are a special categories of plots enabled by Pyleoclim. They allow to plot specific information about a timeseries but are not customizable.

```
pyleoclim.basicSummary(timeseries='', x_axis='', saveFig=False, format='eps', dir='')
```

Makes a basic summary plot

Plots the following information: the time series, location map, Age-Depth profile if both are available from the paleodata, Metadata

Notes

The plots use default settings from the MapLiPD and plotTS methods.

Parameters

- **timeseries** – By default, will prompt for one.
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure.

Examples

```
>>> fig = pyleoclim.basicSummary()
```

```
pyleoclim.basicSummary2(timeseries='', x_axis='', saveFig=False, format='eps', dir='')
```

Second type of basic summary plot

Plots the following information: the time series, a histogram of the PaleoData_values, location map, age-depth profile if both are available from the paleodata, metadata about the record.

Notes

The plots use default settings from the MapLiPD and plotTS methods.

Parameters

- **timeseries** – By default, will prompt for one.
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).

- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure.

Examples

```
>>> fig = pyleoclim.basicSummary2()
```

2.4 Statistics

`pyleoclim.statsTs(timeseries='')`

Calculate the mean and standard deviation of a timeseries

Parameters *timeseries* – sytem will prompt for one if not given

Returns the mean, median, min, max, standard deviation and the inter-quartile range (IQR) of a timeseries.

Examples

```
>>> mean, median, min_, max_, std, IQR = pyleo.statsTs()
0 : Ocean2kHR-AtlanticCapeVerdeMoses2006 : d18O
1 : ODP1098B : SST
2 : ODP1098B : TEX86
3 : Ocean2kHR-AtlanticBahamasTOTORosenheim2005 : d18O
4 : Ocean2kHR-AtlanticBahamasTOTORosenheim2005 : Sr_Ca
5 : Ocean2kHR-PacificClippertonClipp2bWu2014 : Sr_Ca
6 : Crystal.McCabe-Glynn.2013 : sst.anom
7 : Crystal.McCabe-Glynn.2013 : s180carbVPDB
8 : Ocean2kHR-AtlanticMontegoBayHaaseSchramm2003 : d18O
9 : Ocean2kHR-AtlanticMontegoBayHaaseSchramm2003 : Sr_Ca
10 : Ocean2kHR-AtlanticPrincipeSwart1998 : d13C
11 : Ocean2kHR-AtlanticPrincipeSwart1998 : d18O
12 : MD01-2412.Harada.2006 : uk37
13 : MD01-2412.Harada.2006 : sst
14 : MD01-2412.Harada.2006 : calyrbp
15 : MD98-2170.Stott.2004 : mg
16 : MD98-2170.Stott.2004 : RMSE
17 : MD98-2170.Stott.2004 : d18ow
18 : MD98-2170.Stott.2004 : d18o
19 : MD982176.Stott.2004 : d18Ow-s
20 : MD982176.Stott.2004 : sst
21 : MD982176.Stott.2004 : d18Ob.rub
22 : MD982176.Stott.2004 : Mg/Ca-g.rub
23 : Ocean2kHR-PacificNauruGuilderson1999 : d13C
24 : Ocean2kHR-PacificNauruGuilderson1999 : d18O
25 : O2kLR-EmeraldBasin.Sachs.2007 : temperature
26 : O2kLR-EmeraldBasin.Sachs.2007 : notes
27 : O2kLR-EmeraldBasin.Sachs.2007 : Uk37
28 : O2kLR-EmeraldBasin.Sachs.2007 : temperature
29 : O2kLR-EmeraldBasin.Sachs.2007 : notes
30 : O2kLR-EmeraldBasin.Sachs.2007 : Uk37
Enter the number of the variable you wish to use: 13
```



```
>>> print(mean)
10.6708933718
```

```
>>> print(median)
11.0
```

```
>>> print(min_)
5.0
```

```
>>> print(max_)
16.2
```

```
>>> print(std)
2.41519924361
```

```
>>> print(IQR)
3.9
```

`pyleoclim.binTs(timeseries='', x_axis='', bin_size='', start='', end='')`
 Bin the paleoData values of the timeseries

Parameters

- **By default, will prompt the user for one.** (*timeseries.*) –
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **bin_size** (*float*) – the size of the bins to be used. By default, will prompt for one
- **start** (*float*) – Start time/age/depth. Default is the minimum
- **end** (*float*) – End time/age/depth. Default is the maximum

Returns

binned_data- the binned output,
 bins- the bins (centered on the median, i.e. the 100-200 bin is 150),
 n- number of data points in each bin,
 error- the standard error on the mean in each bin

Example

```
>>> ts = pyleoclim.timeseries_list[28]
>>> bin_size = 200
>>> bins, binned_data, n, error = pyleoclim.binTs(timeseries = ts, bin_size = bin_size)
Do you want to plot vs time or depth?
Enter 0 for time and 1 for depth: 0
```

```
>>> print(bins)
[ 239.3   439.3   639.3 ..., 14439.3 14639.3 14839.3]
```

```
>>> print(binned_data)
[28.4400000000000005, 28.9200000000000005, 28.657142857142862,
28.9399999999999998, 28.733333333333334, 28.949999999999999, 28.75,
28.899999999999999, 28.75, 28.566666666666663, 28.800000000000001,
29.049999999999997, 29.233333333333334, 29.274999999999999,
```

```
29.057142857142857, 28.699999999999999, 29.433333333333334,
28.5750000000000003, 28.733333333333331, 28.48, 28.733333333333331,
28.766666666666666, 29.166666666666668, 29.18, 29.600000000000001,
29.3000000000000001, 28.949999999999999, 29.475000000000001,
29.333333333333332, 29.800000000000001, 29.016666666666666,
29.349999999999998, 29.485714285714288, 28.850000000000001,
29.366666666666664, 28.699999999999999, 29.233333333333334,
29.366666666666664, 29.5, 29.350000000000001, 29.699999999999999,
29.3000000000000001, 29.233333333333334, 29.300000000000001,
29.3000000000000001, 29.600000000000001, 28.950000000000003,
29.166666666666668, 28.799999999999997, 28.975000000000001,
29.033333333333331, 28.649999999999999, 28.450000000000003,
28.533333333333331, 28.599999999999998, 28.25, 28.0,
28.550000000000001, 28.799999999999997, 28.350000000000001,
27.699999999999999, 27.149999999999999, 27.666666666666668,
26.800000000000001, 26.700000000000003, 26.800000000000001,
26.5, 26.850000000000001, 26.5, 26.5, 26.0, 26.899999999999999,
26.5, 26.100000000000001]
```

```
pyleoclim.interpTs (timeseries='', x_axis='', interp_step='', start='', end='')
```

Simple linear interpolation

Simple linear interpolation of the data using the numpy.interp method

Parameters

- **Default is blank, will prompt for it** (*timeseries.*) –
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **interp_step** (*float*) – the step size. By default, will prompt the user.
- **start** (*float*) – Start time/age/depth. Default is the minimum
- **end** (*float*) – End time/age/depth. Default is the maximum

Returns

interp_age - the interpolated age/year/depth according to the end/start and time step,

interp_values - the interpolated values

Examples

```
>>> ts = pyleoclim.timeseries_list[28]
>>> interp_step = 200
>>> interp_age, interp_values = pyleoclim.interpTs(timeseries = ts, interp_step = interp_step)
Do you want to plot vs time or depth?
Enter 0 for time and 1 for depth: 0
```

```
>>> print(interp_age)
[ 139.3  339.3  539.3 ..., 14339.3 14539.3 14739.3]
```

```
>>> print(interp_values)
[ 0.188      0.05981567 -0.04020261 ..., 1.20834663 1.47751854
 1.16054494]
```

BASIC MODULE

This module contains methods for basic manipulation of the Paleo/Chron Data.

class `pyleoclim.Basic` (*timeseries_list*)

Basic manipulation of timeseries for scientific purpose.

Calculates statistics of timeseries, bin or interpolate data

static bin_Ts (*timeseries, x_axis='', bin_size='', start='', end=''*)

Bin the PaleoData values

Parameters

- **timeseries** – a single timeseries object. Use `getTSO()` to get one.
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **bin_size** (*float*) – the size of the bins to be used. By default, will prompt for one
- **start** (*float*) – Start time/age/depth. Default is the minimum
- **end** (*float*) – End time/age/depth. Default is the maximum

Returnss: `binned_data` - the binned output

`bins` - the bins (centered on the median, i.e. the 100-200 bin is 150)

`n` - number of data points in each bin

`error` - the standard error on the mean in each bin

static getValues (*timeseries*)

Get the paleoData values from the timeseries object

Parameters **timeseries** – a single timeseries object. Use `getTSO()` to get one from the dictionary

static interp_Ts (*timeseries, x_axis='', interp_step='', start='', end=''*)

Linear interpolation of the PaleoData values

Parameters

- **timeseries** – a timeseries object. Use `getTSO()` to get one.
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **interp_step** (*float*) – the step size. By default, will prompt the user.

- **start** (*float*) – Start time/age/depth. Default is the minimum
- **end** (*float*) – End time/age/depth. Default is the maximum

Returns

interp_age - the interpolated age/year/depth according to the end/start and time step

interp_values - the interpolated values

static onCommonXAxis (*timeseries1, timeseries2, x_axis='', interp_step='', start='', end=''*)

Places two timeseries on a common x-axis

Interpolate the PaleoDataValues of two timeseries on a common x-axis.

Parameters

- **timeseries1** – a timeseries object. Use getTSO() to get one.
- **timeseries2** – a timeseries object. Use getTSO() to get one.
- **x-axis** (*str*) – The representation against which to express the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **interp_step** (*float*) – the step size. By default, will prompt the user.
- **start** (*float*) – Start time/age/depth. Default is the maximum of the minima of the two timeseries
- **end** (*float*) – End time/age/depth. Default is the minimum of the maxima of the two timeseries

Returns

interp_age - the interpolated age/year/depth according to the end/start and time step

interp_values1 - the interpolated values for the first timeseries

interp_values2 - the interpolated values for the second timeseries

simpleStats (*timeseries=''*)

Compute the mean and standard deviation of a time series

Parameters **timeseries** – a single timeseries. Will prompt for one if not available

Returns the mean, median, min, max, standard deviation and IQR of a timeseries.

MAPPING MODULE

class `pyleoclim.Map` (*plot_default*)
Create Maps using Basemap.

Uses the default color palette: `pyleoclim.plot_default`

map_Lipd (*name*='', *countries*=True, *counties*=False, *rivers*=False, *states*=False, *background*='shadedrelief', *scale*=0.5, *markersize*=50, *marker*='default', *saveFig*=False, *dir*='', *format*='eps')
Makes a map for a single record.

Parameters

- **name** (*str*) – the name of the LiPD file. **WITH THE .LPD EXTENSION!**. If not provided, will prompt the user for one.
- **countries** (*bool*) – Draws the country borders. Default is on (True).
- **counties** (*bool*) – Draws the USA counties. Default is off (False).
- **states** (*bool*) – Draws the American and Australian states borders. Default is off (False).
- **background** (*str*) – Plots one of the following images on the map: `bluemarble`, `etopo`, `shadedrelief`, or `none` (filled continents). Default is `shadedrelief`.
- **scale** (*float*) – useful to downgrade the original image resolution to speed up the process. Default is 0.5.
- **markersize** (*int*) – default is 100
- **marker** (*str*) – a string (or list) containing the color and shape of the marker. Default is by `archiveType`. Type `pyleo.plot_default` to see the default palette.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called 'figures' in the LiPD working directory (`lipd.path`).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is "eps". Most backend support `png`, `pdf`, `ps`, `eps`, and `svg`.

Returns The Figure

map_all (*markersize*=50, *saveFig*=False, *dir*='', *format*='eps')
Map all the available records loaded into the LiPD working directory by `archiveType`.

Parameters

- **markersize** (*int*) – default is 50
- **saveFig** (*bool*) – default is to save the figure

- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure

TIMESERIES PLOTS MODULE

class `pyleoclim.Plot` (*plot_default, timeseries_list*)
 Plot a timeseries

plot_Ts (*timeseries='', x_axis='', markersize=50, marker='default', saveFig=False, dir='', format='eps'*)
 Plot a timeseries object

Parameters

- **timeseries** – A timeseries. By default, will prompt the user for one.
- **x_axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **markersize** (*int*) – default is 50.
- **marker** (*str*) – a string (or list) containing the color and shape of the marker. Default is by `archiveType`. Type `pyleo.plot_default` to see the default palette.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (`lipd.path`).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure

plot_agemodel (*timeseries='', markersize=50, marker='default', saveFig=True, dir='', format='eps'*)
 Make a simple age-depth profile

Parameters

- **timeseries** – A timeseries. By default, will prompt the user for one.
- **markersize** (*int*) – default is 50.
- **marker** (*str*) – a string (or list) containing the color and shape of the marker. Default is by `archiveType`. Type `pyleo.plot_default` to see the default palette.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (`lipd.path`).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure

```
plot_hist (timeseries='', bins=None, hist=True, kde=True, rug=False, fit=None, hist_kws={'label':  
    'hist'}, kde_kws={'label': 'kde'}, rug_kws={'label': 'rug'}, fit_kws={'label': 'fit'},  
    color=None, vertical=False, norm_hist=True, legend=True, saveFig=False, format='eps',  
    dir='')
```

Plot a univariate distribution of the PaleoData values

This function is based on the seaborn displot function, which is itself a combination of the matplotlib hist function with the seaborn kdeplot() and rugplot() functions. It can also fit scipy.stats distributions and plot the estimated PDF over the data.

Parameters

- **timeseries** – A timeseries. By default, will prompt the user for one.
- **bins** (*int*) – Specification of hist bins following matplotlib(hist), or None to use Freedman-Diaconis rule
- **hist** (*bool*) – Whether to plot a (normed) histogram
- **kde** (*bool*) – Whether to plot a gaussian kernel density estimate
- **rug** (*bool*) – Whether to draw a rugplot on the support axis
- **fit** – Random variable object. An object with fit method, returning a tuple that can be passed to a pdf method of positional arguments following a grid of values to evaluate the pdf on.
- **kde, rug, fit}_kws** (*{hist,}*) – Dictionaries. Keyword arguments for underlying plotting functions. If modifying the dictionary, make sure the labels “hist”, “kde”, “rug” and “fit” are still passed.
- **color** (*str*) – matplotlib color. Color to plot everything but the fitted curve in.
- **vertical** (*bool*) – if True, observed values are on y-axis.
- **norm_hist** (*bool*) – If True (default), the histogram height shows a density rather than a count. This is implied if a KDE or fitted density is plotted
- **legend** (*bool*) – If true, plots a default legend
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns fig - The figure

SUMMARY PLOTS MODULE

class `pyleoclim.SummaryPlots` (*timeseries_list*, *plot_default*)

Plots various summary figures for a LiPD record

TsData (*timeseries*='', *x_axis*='')

Get the PaleoData with age/depth information

Get the necessary information for the TS plots/necessary to allow for axes specification

Parameters

- **timeseries** – a single timeseries object. By default, will prompt the user
- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.

Returns

`dataframe` - a dataframe containing the x- and y-values

`archiveType` - the archiveType (for plot settings)

`x_axis_label` - the label for the x-axis

`y_axis_label` - the label for the y-axis

`headers` - the headers of the dataframe

agemodelData (*timeseries*='')

Get the necessary information for the agemodel plot

Parameters **timeseries** – a single timeseries object. By default, will prompt the user

Returns

`depth` - the depth values

`age` - the age values

`x_axis_label` - the label for the x-axis

`y_axis_label` - the label for the y-axis

`archiveType` - the archiveType (for default plot settings)

basic (*x_axis*='', *timeseries*='', *saveFig*=False, *format*='eps', *dir*='')

Makes a basic summary plot

Plots the following information: the time series, location map, Age-Depth profile if both are available from the paleodata, Metadata

Notes

The plots use default settings from the MapLiPD and plotTS methods.

Parameters

- **timeseries** – By default, will prompt for one.
- **x_axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure.

basic2 (*x_axis='', timeseries='', saveFig=False, format='eps', dir=''*)

Second type of basic summary plot

Plots the following information: the time series, a histogram of the PaleoData_values, location map, age-depth profile if both are available from the paleodata, metadata about the record.

Parameters

- **timeseries** – a timeseries object. By default, will prompt for one
- **x_axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.
- **saveFig** (*bool*) – default is to not save the figure
- **dir** (*str*) – the full path of the directory in which to save the figure. If not provided, creates a default folder called ‘figures’ in the LiPD working directory (lipd.path).
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.

Returns The figure

getMetadata (*timeseries*)

Get the necessary metadata to be printed out automatically

Parameters **timeseries** – a specific timeseries object

Returns

archiveType

Authors (if more than 2, replace by et al.

PublicationYear

Publication DOI

Variable Name

Units

Climate Interpretation

Calibration Equation

Calibration References

Calibration Notes

Return type A dictionary containing the following metadata

MANIPULATING LIPD FILES

The following methods allows to manipulate LiPD files and objects.

7.1 Creating directories and saving figures

`pyleoclim.createDir(path, foldername)`

Create a new folder in a working directory

Create a new folder in a working directory to save outputs from Pyleoclim.

Parameters

- **path** (*str*) – the path to the new folder.
- **foldername** (*str*) – the name of the folder to be created

Returns newdir - the full path to the new directory

`pyleoclim.saveFigure(name, format='eps', dir='')`

Save a figure

Save the figure in the directory. If not given, creates a folder in the lipd.path directory.

Parameters

- **name** (*str*) – name of the file
- **format** (*str*) – One of the file extensions supported by the active backend. Default is “eps”. Most backend support png, pdf, ps, eps, and svg.
- **dir** (*str*) – the name of the folder in the LiPD working directory. If not provided, creates a default folder called ‘figures’.

7.2 Manipulating LiPD files

`pyleoclim.enumerateLipds()`

Enumerate the LiPD files loaded in the workspace

`pyleoclim.promptForLipd()`

Prompt for a LiPD file

Ask the user to select a LiPD file from a list Use this function in conjunction with enumerateLipds()

Returns The index of the LiPD file

7.3 Manipulating Variables in a LiPD file

`pyleoclim.promptForVariable()`

Prompt for a specific variable

Ask the user to select the variable they are interested in. Use this function in conjunction with `readHeaders()` or `getTSO()`

Returns The index of the variable

`pyleoclim.valuesLoc(dataframe, missing_value='NaN', var_idx=1)`

Remove missing values flag

Look for the indexes where there are no missing values for the variable

Parameters

- **dataframe** – a Pandas Dataframe
- **missing_value** (*str or float*) – how are the missing value represented. Default is NaN
- **var_idx** (*int*) – the column number in which to look for the missing values (default is the second column)

Returns `val_idx` - the indices of the lines in the dataframe containing the actual values

`pyleoclim.xAxisTs(timeseries)`

Prompt the user to choose a x-axis representation for the timeseries.

Parameters **timeseries** – a timeseries object

Returns

- `x_axis` - the values for the x-axis representation,
- `label` - returns either “age”, “year”, or “depth”

7.4 Manipulating timeseries objects

`pyleoclim.enumerateTs(timeseries_list)`

Enumerate the available time series objects

Parameters **timeseries_list** – a list of available timeseries objects. To use the timeseries loaded upon initiation of the pyleoclim package, use `pyleo.time_series`.

`pyleoclim.getTs(timeseries_list)`

Get a specific timeseries object from a dictionary of timeseries

Parameters **timeseries_list** – a list of available timeseries objects. To use the timeseries loaded upon initiation of the pyleoclim package, use `pyleo.time_series`.

Returns A single timeseries object

`pyleoclim.TsToDf(timeseries, x_axis='')`

Timeseries to Dataframe

Create a dataframe from a timeseries object with two colums: depth/age representation and the paleoData values

Parameters

- **timeseries** – A timeseries object

- **x-axis** (*str*) – The representation against which to plot the paleo-data. Options are “age”, “year”, and “depth”. Default is to let the system choose if only one available or prompt the user.

Returns A Pandas Dataframe with two columns - the x-axis representation (“year”, “age”, or “depth”) and the PaleoDataValues

7.5 Handling mapping to LinkedEarth Ontology

`pyleoclim.LipdToOntology` (*archiveType*)
standardize archiveType

Transform the archiveType from their LiPD name to their ontology counterpart

Parameters **archiveType** (*STR*) – name of the archiveType from the LiPD file

Returns archiveType according to the ontology

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

A

agemodelData() (pyleoclim.SummaryPlots method), 21

B

Basic (class in pyleoclim), 15

basic() (pyleoclim.SummaryPlots method), 21

basic2() (pyleoclim.SummaryPlots method), 22

basicSummary() (in module pyleoclim), 11

basicSummary2() (in module pyleoclim), 11

bin_Ts() (pyleoclim.Basic static method), 15

binTs() (in module pyleoclim), 13

C

createDir() (in module pyleoclim), 25

E

enumerateLipds() (in module pyleoclim), 25

enumerateTs() (in module pyleoclim), 26

G

getMetadata() (pyleoclim.SummaryPlots method), 22

getTs() (in module pyleoclim), 26

getValues() (pyleoclim.Basic static method), 15

H

histTs() (in module pyleoclim), 10

I

interp_Ts() (pyleoclim.Basic static method), 15

interpTs() (in module pyleoclim), 14

L

LipdToOntology() (in module pyleoclim), 27

M

Map (class in pyleoclim), 17

map_all() (pyleoclim.Map method), 17

map_Lipd() (pyleoclim.Map method), 17

mapAll() (in module pyleoclim), 8

mapLipd() (in module pyleoclim), 8

O

onCommonXAxis() (pyleoclim.Basic static method), 16

openLipds() (in module pyleoclim), 7

P

Plot (class in pyleoclim), 19

plot_agemodel() (pyleoclim.Plot method), 19

plot_hist() (pyleoclim.Plot method), 20

plot_Ts() (pyleoclim.Plot method), 19

plotTs() (in module pyleoclim), 9

promptForLipd() (in module pyleoclim), 25

promptForVariable() (in module pyleoclim), 26

S

saveFigure() (in module pyleoclim), 25

simpleStats() (pyleoclim.Basic method), 16

statsTs() (in module pyleoclim), 12

SummaryPlots (class in pyleoclim), 21

T

TsData() (pyleoclim.SummaryPlots method), 21

TsToDf() (in module pyleoclim), 26

V

valuesLoc() (in module pyleoclim), 26

X

xAxisTs() (in module pyleoclim), 26