divebomb Documentation

Release 1.1.0

Alex Nunes

Jun 07, 2019

Contents:

Divebomb is a python package that uses pandas to divide a timeseries of depths into individual dives. The dives are profiled as a `Dive` or `DeepDive` depending on the animal. The `Dive` class is used for frequently surfacing animals, such as seals and whales. The `DeepDive` class is used for infrequently surfaceing animals, like sharks.

The dive profiles are reduced to 8 dimensions using Principal Component Analsysis. Guassian Mixed Models are generated using theses variables and the minimal Bayesian Information Criterion is used to determine the optimal number of clusters. The dives are split into the clusters using Agglomerative Hierarchical Clustering (from sklearn). The dives are then display through iPython notebooks or saved to netCDF files organized by cluster.

CHAPTER 1

Dive

A `Dive` is then profiled with the following attributes:

- `max_depth` - the max depth in the dive
- `dive_start` - the timestamp of the first point in the dive
- `dive_end` - the timestamp of the last point in the dive
- `bottom_start` - the timestamp of the first point in the dive when the animal is at depth
- `td_bottom_duration` - a timedelta object containing the duration of the time the animal is at depth in seconds
- `td_descent_duration` - a timedelta object containing the duration of the time the animal is descending in seconds
- `td_ascent_duration` - a timedelta object containing the duration of the time the animal is ascending in seconds
- `td_surface_duration` - a timedelta object containing the duration of the time the animal is at the surface in seconds
- `bottom_variance` - the variance of the depth while the animal is at the bottom of the dive
- `dive_variance` - the variance of the depth for the entire dive.
- `descent_velocity` - the average velocity of the descent
- `ascent_velocity` - the average velocity of the descent
- `peaks` - the number of peaks found in the dive profile
- `left_skew` - a boolean of 1 or 0 indicating if the dive is left skewed
- `right_skew` - a boolean of 1 or 0 indicating if the dive is right skewed
- `no_skew` - a boolean of 1 or 0 indicating if the dive is not skewed

CHAPTER 2

DeepDive

A `DeepDive` is then profiled with the following attributes:

- `max_depth` - the max depth in the dive
- `min_depth` - the max depth in the dive
- `dive_start` - the timestamp of the first point in the dive
- `dive_end` - the timestamp of the last point in the dive
- `td_total_duration` - a timedelta (in seconds since 1970-01-01) containing the duration of the dive
- `depth_variance` - the variance of the depth for the entire dive.
- `average_vertical_velocity` - the mean velocity of the animal over the entire dive with negative value indicating upward movement
- `average_descent_velocity` - the average velocity of any downward movement as positive value
- `average_ascent_velocity` - the average velocity of any upward movement as positive value
- `number_of_descent_transitions` - the number of times and animal moves descends any distance in the dive period
- `number_of_ascent_transitions` - the number of times and animal moves ascends any distance in the dive period
- `total_descent_distance_traveled` - the total absolute distance in meters in which the anaimal moves down
- `total_ascent_distance_traveled` - the total absolute distance in meters in which the anaimal moves up
- `overall_change_in_depth` - the difference between the minimum and maximum depth within the dive period
- `td_time_at_depth` - the duration in seconds at which the animal spends in the deepest part of the vertical movement (< 85% depth)

- `td_time_pre_depth` - the duration in seconds befor the deepest part of the vertical movement (< 85% depth)
- `td_time_post_depth` - the duration in seconds after the deepest part of the vertical movement (< 85% depth)
- `peaks` - the number of peaks found in the dive profile
- `left_skew` - a boolean of 1 or 0 indicating if the dive is left skewed
- `right_skew` - a boolean of 1 or 0 indicating if the dive is right skewed
- `no_skew` - a boolean of 1 or 0 indicating if the dive is not skewed

CHAPTER 3

Surface Thresholds

A surface threshold is used for surfacing animals to define a depth window for what is considered to be at surface. The `surface_threshold` argument defaults to `0` but can be changed in the `profile_dives()` function. For example `surface_threshold=2` might be passed for animal that is `~2` meters long. `surface_threshold` is always passed in meters.

CHAPTER 4

At Depth Thresholds

An at depth threshold is used in both the `Dive` and the `DeepDive` class. The `at_depth_thresold` argument is a value between `0` and `1` that determines the window for when an animal is considered to be at bottom of its dive. The default value is `0.15` which means the bottom `15%` of the relative depth is considered to be at bottom. `at_depth_thresold` is always as value between `0` and `1` expressing a percentage.

CHAPTER 5

Dive Detection

There are two arguments that are used to help determine dives on any animal, `dive_detection_sensitivity` and `minimal_time_between_dives`. The `dive_detection_sensitivity` argument is a value between `0` and `1`. The default is `0.98` for surfacing animals and `0.5` for non-surfacing animals. The `dive_detection_sensitivity` helps determine range where dive starts can be determined. The `minimal_time_between_dives` is the minimum time (in seconds) that has to occur before a new dive can start. The default value for this is `10` seconds.

CHAPTER 6

Skews

A skew is defined as any difference one way or the other in descent or ascent times for the `Dive` class and any difference in pre-depth or post-depth time for a `DeepDive`. This method was chosen as researchers found skew was most accurately represented when any difference between the values existed.

CHAPTER 7

Timestamps

The input timestamps are expected to be in a datetime format. The output timestamps are in `seconds since 1970-01-01`. Every netCDF file has the time unit saved as an attribute as a reminder. All dive attributes that start with `td_` are a duration in seconds. The `time`, `dive_start`, `dive_end`, and `bottom_start` will use the units mentioned above. the netCDF4 library has a `num2date` function that will convert the values back to a datetime object.

7.1 Divebomb

Divebomb is a python package that uses pandas to divide a timeseries of depths into individual dives. The dives are profiled as a `Dive` or `DeepDive` depending on the animal. The `Dive` class is used for frequently surfacing animals, such as seals and whales. The `DeepDive` class is used for infrequently surfaceing animals, like sharks.

The dive profiles are reduced to 8 dimensions using Principal Component Analsysis. Guassian Mixed Models are generated using theses variables and the minimal Bayesian Information Criterion is used to determine the optimal number of clusters. The dives are split into the clusters using Agglomerative Hierarchical Clustering (from sklearn). The dives are then display through iPython notebooks or saved to netCDF files organized by cluster.

7.1.1 Dive

A `Dive` is then profiled with the following attributes:

- `max_depth` - the max depth in the dive
- `dive_start` - the timestamp of the first point in the dive
- `dive_end` - the timestamp of the last point in the dive
- `bottom_start` - the timestamp of the first point in the dive when the animal is at depth
- `td_bottom_duration` - a timedelta object containing the duration of the time the animal is at depth in seconds

- `td_descent_duration` - a timedelta object containing the duration of the time the animal is descending in seconds
- `td_ascent_duration` - a timedelta object containing the duration of the time the animal is ascending in seconds
- `td_surface_duration` - a timedelta object containing the duration of the time the animal is at the surface in seconds
- `bottom_variance` - the variance of the depth while the animal is at the bottom of the dive
- `dive_variance` - the variance of the depth for the entire dive.
- `descent_velocity` - the average velocity of the descent
- `ascent_velocity` - the average velocity of the descent
- `peaks` - the number of peaks found in the dive profile
- `left_skew` - a boolean of 1 or 0 indicating if the dive is left skewed
- `right_skew` - a boolean of 1 or 0 indicating if the dive is right skewed
- `no_skew` - a boolean of 1 or 0 indicating if the dive is not skewed

7.1.2 DeepDive

A `DeepDive` is then profiled with the following attributes:

- `max_depth` - the max depth in the dive
- `min_depth` - the max depth in the dive
- `dive_start` - the timestamp of the first point in the dive
- `dive_end` - the timestamp of the last point in the dive
- `td_total_duration` - a timedelta (in seconds since 1970-01-01) containing the duration of the dive
- `depth_variance` - the variance of the depth for the entire dive.
- `average_vertical_velocity` - the mean velocity of the animal over the entire dive with negative value indicating upward movement
- `average_descent_velocity` - the average velocity of any downward movement as positive value
- `average_ascent_velocity` - the average velocity of any upward movement as positive value
- `number_of_descent_transitions` - the number of times and animal moves descends any distance in the dive period
- `number_of_ascent_transitions` - the number of times and animal moves ascends any distance in the dive period
- `total_descent_distance_traveled` - the total absolute distance in meters in which the anaimal moves down
- `total_ascent_distance_traveled` - the total absolute distance in meters in which the anaimal moves up
- `overall_change_in_depth` - the difference between the minimum and maximum depth within the dive period
- `td_time_at_depth` - the duration in seconds at which the animal spends in the deepest part of the vertical movement (< 85% depth)

- `td_time_pre_depth` - the duration in seconds befor the deepest part of the vertical movement (< 85% depth)
- `td_time_post_depth` - the duration in seconds after the deepest part of the vertical movement (< 85% depth)
- `peaks` - the number of peaks found in the dive profile
- `left_skew` - a boolean of 1 or 0 indicating if the dive is left skewed
- `right_skew` - a boolean of 1 or 0 indicating if the dive is right skewed
- `no_skew` - a boolean of 1 or 0 indicating if the dive is not skewed

7.1.3 Surface Thresholds

A surface threshold is used for surfacing animals to define a depth window for what is considered to be at surface. The `surface_threshold` argument defaults to `0` but can be changed in the `profile_dives()` function. For example `surface_threshold=2` might be passed for animal that is `~2` meters long. `surface_threshold` is always passed in meters.

7.1.4 At Depth Thresholds

An at depth threshold is used in both the `Dive` and the `DeepDive` class. The `at_depth_thresold` argument is a value between `0` and `1` that determines the window for when an animal is considered to be at bottom of its dive. The default value is `0.15` which means the bottom `15%` of the relative depth is considered to be at bottom. `at_depth_thresold` is always as value between `0` and `1` expressing a percentage.

7.1.5 Dive Detection

There are two arguments that are used to help determine dives on any animal, `dive_detection_sensitivity` and `minimal_time_between_dives`. The `dive_detection_sensitivity` argument is a value between `0` and `1`. The default is `0.98` for surfacing animals and `0.5` for non-surfacing animals. The `dive_detection_sensitivity` helps determine range where dive starts can be determined. The `minimal_time_between_dives` is the minimum time (in seconds) that has to occur before a new dive can start. The default value for this is `10` seconds.

7.1.6 Skews

A skew is defined as any difference one way or the other in descent or ascent times for the `Dive` class and any difference in pre-depth or post-depth time for a `DeepDive`. This method was chosen as researchers found skew was most accurately represented when any difference between the values existed.

7.1.7 Timestamps

The input timestamps are expected to be in a datetime format. The output timestamps are in `seconds since 1970-01-01`. Every netCDF file has the time unit saved as an attribute as a reminder. All dive attributes that start with `td_` are a duration in seconds. The `time`, `dive_start`, `dive_end`, and `bottom_start` will use the units mentioned above. the netCDF4 library has a `num2date` function that will convert the values back to a datetime object.

7.2 Installation

Divebomb can be installed using Pip or through a Conda environment.

7.2.1 Conda

```
conda config --add channels conda-forge
conda install divebomb
```

7.2.2 Pip

```
pip install divebomb
```

7.3 Code Examples

7.3.1 Divebomb

The example data set below is dive data from grey seal over the course of a few days.

Example data set: `Seal Dives`

Dives

Pass a Pandas DataFrame to the function with a `time` and a `depth` (in positive meters) column. Provide the surface threshold using `surface_threshold` (in meters). Refine other arguments as needed.

```
from divebomb import profile_cluster_export
import pandas as pd

data = pd.read_csv('/path/to/data.csv')
surface_threshold=3

profile_cluster_export(data, folder='results', surface_threshold=surface_threshold ,␣
↪columns={'depth': 'depth', 'time': 'time'})
```

DeepDives

To run the `profile_cluster_export()` function on an animal, such as a shark, just set `is_surfacing_animal==False`. This variable makes the function call the `DeepDive` class instead. `DeepDives` are not dependent on the animal surfacing again.

```
import pandas as pd
from divebomb import profile_cluster_export

df = pd.read_csv('/path/to/data.csv')

dives = profile_cluster_export(df, folder='results', is_surfacing_animal=False)
```

Changing Surface threshold

A surface threshold is used for surfacing animals to define a depth window for what is considered to be at surface. The `surface_threshold` argument defaults to `0` but can be changed in the `profile_cluster_export()` function. For example `surface_threshold=2` might be passed for animal that is `~2` meters long. `surface_threshold` is always passed in meters.

Example:

```
import pandas as pd
from divebomb import profile_cluster_export

data = pd.read_csv('data.csv')

surface_threshold = 3 # in meters

dives = profile_cluster_export(data, folder='results', surface_threshold=surface_
↪threshold)
```

Changing At Depth Threshold

An at depth threshold is used in both the `Dive` and the `DeepDive` class. The `at_depth_thresold` argument is a value between `0` and `1` that determines the window for when an animal is considered to be at bottom of its dive. The default value is `0.15` which means the bottom `15%` of the relative depth is considered to be at bottom. `at_depth_thresold` is always as value between `0` and `1` expressing a percentage.

Example:

```
import pandas as pd
from divebomb import profile_cluster_export

data = pd.read_csv('data.csv')

at_depth_threshold = 0.2 # A value betwen 0 and 1

dives = profile_cluster_export(data, folder='results', minimal_time_between_
↪dives=minimal_time_between_dives)
```

Changing Dive Detection Sensitivity

The `dive_detection_sensitivity` argument is a value between `0` and `1`. The default is `0.98` for surfacing animals and `0.5` for non-surfacing animals. The `dive_detection_sensitivity` helps determine range where dive starts can be determined.

Example:

```
import pandas as pd
from divebomb import profile_cluster_export

data = pd.read_csv('data.csv')

dive_detection_sensitivity = 0.95

dives = profile_cluster_export(data, folder='results', dive_detection_
↪sensitivity=dive_detection_sensitivity)
```

Changing Minimal Time Between Dives

The `minimal_time_between_dives` is the minimum time (in seconds) that has to occur before a new dive can start. The default value for this is `10` seconds.

Example:

```
import pandas as pd
from divebomb import profile_cluster_export

data = pd.read_csv('data.csv')

minimal_time_between_dives = 600 # in seconds

dives = profile_cluster_export(data, folder='results', minimal_time_between_
↪dives=minimal_time_between_dives)
```

7.3.2 Separating Out Components

Each of the components from *profile_cluster_export()* can run separately but their input may rely on the out put from the previous. Below is how to run each of the components separately to modify the clustering or export to CSVs

Profile Dives

The `profile_dives()` function only profiles the dives. It finds the start points for the dives, then finds the dive attributes. `profile_dives()` takes the `surface_threshold`, `dive_detection_sensitivity`, `at_depth_thresold`, and `is_surfacing_animal` arguments just like `profile_cluster_export()`. It returns three datasets of the profiled dives, any insufficient dives, and the original data.

```
from divebomb import profile_dives
import pandas as pd

data = pd.read_csv('/path/to/data.csv')
surface_threshold=3

# Profile dives and save the 3 outputs
dives, insufficient_dives, data = profile_dives(data, surface_threshold=surface_
↪threshold)
```

`profile_dives()` also takes and argument to display the dive in a Jupyter Notebook. If `ipython_display_mode=True` then the dives will be displayed with with a slider to choose the dive.

```
from divebomb import profile_dives
import pandas as pd

data = pd.read_csv('/path/to/data.csv')
surface_threshold=3

profile_dives(data, surface_threshold=surface_threshold, ipython_display_mode=True)
```

Cluster Dives

The `cluster_dives()` functions will take a DataFrame of profiled dives and cluster on the arguments passed. You can adjust the number of clusters, the principle component analysis (PCA) components, and which attributes are used

througharguments in the function. `cluster_dives()` returns three datasets: the dives with cluster number, the loadings matrix for the PCA, and the PCA matrix. Below are some examples.

```
from divebomb import profile_dives, cluster_dives
import pandas as pd

data = pd.read_csv('/path/to/data.csv')
surface_threshold=3

dives, insufficient_dives, data = profile_dives(data, surface_threshold=surface_
↪threshold)

# Get the profiled dives from the profile_dives function above and
# assign the 3 datasets to variables
clustered_dives, loadings, pca_output_matrix = cluster_dives(dives)
```

Below is an example of overriding the number of clusters generated.

```
clustered_dives, loadings, pca_output_matrix  = cluster_dives(dives, n_cluster=4)
```

Below is an example of overriding dimensionality reduction in the PCA (the default is 8). `pca_components` must be less than or equal to the number of columns/attributes being used for the clustering (`dive_start`, `dive_end`, `surface_threshold`, and `insufficient_data` will not count towards the number of columns/attributes).

```
clustered_dives, loadings, pca_output_matrix  = cluster_dives(dives, pca_components=4)
```

Below is an example of selecting which attributes are used in the clustering. The code only clusters on `td_ascent_duration`, `td_bottom_duration`, `td_descent_duration`, and `td_dive_duration`. We choose `pca_components=2` to reduce the dimensionality from 4 to 2.

```
clustered_dives, loadings, pca_output_matrix = cluster_dives(dives,
                                                             pca_components=2,
                                                             attributes=['td_ascent_
↪duration',
                                                                         'td_bottom_
↪duration',
                                                                         'td_descent_
↪duration',
                                                                         'td_dive_
↪duration'])
```

Export Dives

Dives can either be exported to NetCDF or CSV. Both `profile_dives()` and `cluster_dives()` need to be run and assigned to variables to get all dataset created in the process.

`export_to_netcdf()` will take all of the datasets and save them to a `.nc` file as well as saving a `.nc` for each individual dive in folders sorted by cluster.

```
from divebomb import profile_dives, cluster_dives, export_to_csv, export_to_netcdf
import pandas as pd

data = pd.read_csv('/path/to/data.csv')
surface_threshold=3

dives, insufficient_dives, data = profile_dives(data, surface_threshold=surface_
↪threshold)
```

(continues on next page)

(continued from previous page)

```
# Get the profiled dives from the profile_dives function above
clustered_dives, loadings, pca_output_matrix s = cluster_dives(dives)

# Export to netcdf
export_to_netcdf(folder = "nc_results",
                 data = data,
                 dives=clustered_dives,
                 loadings=loadings,
                 pca_output_matrix=pca_output_matrix,
                 insufficient_dives=insufficient_dives)
```

`export_to_csv` will take the inputs and save the clustered dives, loadings, and PCA matrix to a folder as CSVs.

```
# Export to CSV (no individual dive files)
export_to_csv(folder = "csv_results",
              dives=clustered_dives,
              loadings=loadings,
              pca_output_matrix=pca_output_matrix,
              insufficient_dives=insufficient_dives)
```

All outputs are DataFrames and can be saved individually by appending `.to_csv('filename.csv', index=False)` to the variable. For example, the code below will save the profiled dives (no clustering) to a CSV.

```
from divebomb import profile_dives
import pandas as pd

data = pd.read_csv('/path/to/data.csv')
surface_threshold=3

# Profile dives and save the 3 outputs
dives, insufficient_dives, data = profile_dives(data, surface_threshold=surface_
↪threshold)

dives.to_csv('profile_dives.csv', index=False)
```

7.3.3 Plotting Results

Divebomb includes two functions to plot dives. The first, `plot_from_nc()` will plot a single dive with disinguished phases. `plot_from_nc()` includes a `type` argument that can either be `dive` or `deepdive`.

The second function `cluster_summary_plot` will plot the minimum, maximum, and mean depth for each cluster. Time is asjusted to be the number of seconds into the dive, rather than a timestamp. Both axes can be individually scaled relative to maximum values of the clusters. For example, time can be scaled to be a proigress percentage through the dive. Scaling can be applied by passing the following: `scale={'depth'=True, 'time':True}` Below are examples and how they can be applied.

Single Dive

Below is an example of a single dive from a surfacing animal.

```
from divebomb.plotting import plot_from_nc, cluster_summary_plot
```

(continues on next page)

(continued from previous page)

```
path = '/path/to/results_folder'
cluster = 2
dive_id = 555

# Plot inside a notebook
plot_from_nc(path, cluster, dive_id, ipython_display=True)

# Plot out to an HTML file
plot_from_nc(path, cluster, dive_id, ipython_display=False, filename="dive.html")
```

Dive Clusters

Below is an example of the clusters from a surfacing animal.

```
from divebomb.plotting import cluster_summary_plot

path = '/path/to/results_folder'

# Plot inside a notebook
cluster_summary_plot(path, ipython_display=True)

# Plot out to an HTML file
cluster_summary_plot(path, ipython_display=False, filename="clusters.html", scale={
↪'depth':False, 'time':True})
```

Single DeepDive

Below is an example of non-surfacing animal dive. This example is also a sparser dataset as there are 10 minutes between data points.

```
from divebomb.plotting import plot_from_nc, cluster_summary_plot

path = '/path/to/results_folder'
cluster = 3
dive_id = 68

# Plot inside a notebook
plot_from_nc(path, cluster, dive_id, ipython_display=True, type='deepdive)

# Plot out to an HTML file
plot_from_nc(path, cluster, dive_id, ipython_display=False, filename='single_deepdive.
↪html', type='deepdive')
```

Clustered DeepDives

Below is an example of the clusters from a non-surfacing animal. This example is also a sparser dataset as there are 10 minutes between data points.

```
from divebomb.plotting import cluster_summary_plot

path = '/path/to/results_folder'
```

(continues on next page)

(continued from previous page)

```
# Plot inside a notebook
cluster_summary_plot(path, ipython_display=True)

# Plot out to an HTML file
cluster_summary_plot(path, ipython_display=False, filename='deepdive_clusters.html',␣
↪title='DeepDive Clusters')
```

7.3.4 Correcting Depth on Surfacing Animals

Depth recordings can be uncalihrated or drift over time. The following are two ways from divebomb's *preprocessing module* to correct for the offset on a **surfacing animal**. The data passes to the function must have `time` and a `depth` (in positive meters) columns. The first uses a local max:

```
from divebomb import profile_cluster_export
import pandas as pd
window = 3600 #seconds

data = pd.read_csv('/path/to/data.csv')
corrected_depth_data = correct_depth_offset(data, window=window, aux_file='results/
↪aux_file.nc')
```

The second wethod uses a rolling average of all surface and near surface values in the time window:

```
from divebomb import profile_cluster_export
import pandas as pd
window = 3600 # seconds
surface_threshold = 4 # meters

data = pd.read_csv('/path/to/data.csv')
corrected_depth_data = correct_depth_offset(data, window=window, method='mean',␣
↪surface_threshold=surface_threshold, aux_file='results/aux_file.nc')
```

7.4 Divebomb Functions

The following are the primary functions by divebomb to process the dives. The main function is `profile_dives()` and the other functions (`display_dive()`, `cluster_dives()`, and `export_dives()`) are used has helper functions inside `profile_dives()`.

`divebomb.`**`clean_dive_data`**(*data*, *columns={'depth': 'depth', 'time': 'time'}*)

Parameters

- **`data`** – a Pandas DataFrame consisting of a time and a depth column
- **`columns`** – column renaming dictionary if needed

Returns a Pandas DataFrame with `time` in seconds since 1970-10-01 and `depth`

`divebomb.`**`cluster_dives`**(*dives*, *pca_components=8*, *n_clusters=None*, *attributes=None*)

This function takes advantage of sklearn and reduces the dimensionality with Principal Component Analysis, finds the optimal number of n_clusters using Gaussian Mixed Models and the Bayesion Information Criterion, then uses Agglomerative Clustering on the dives profiles to group them.

Parameters

- **dives** – a pandas DataFrame of dive attributes
- **pca_components** – the number of components for dimensionality reduction. Should be fewer than the number of columns in the dataset.
- **n_clusters** – An override for the number of clusters to find when clustering
- **attributes** – A list of variable/columns to use during the process. This can be a subset of the columns in the data.

Returns the clustered dives, the PCA loadings matrix, and the PCA output matrix

divebomb.**display_dive**(*index*, *data*, *starts*, *type='dive'*, *surface_threshold=0*, *at_depth_threshold=0.15*)

This function just takes the index, the data, and the starts and displays the dive using plotly. It is used as a helper method for viewing the dives if `ipython_display` is `True` in `profile_dives()`.

Parameters

- **index** – the index of the dive profile to plot
- **data** – the dataframe of the original dive data
- **starts** – the dataframe of the dive starts
- **type** – s tring that indicates using either the `Dive` or `DeepDive` class
- **surface_threshold** – the calculated surface threshold based on animal length
- **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns a dive plot from plotly

divebomb.**export_dives**(*dives*, *data*, *folder*, *is_surface_events=False*)

This function exports each dive to its own netCDF file grouped by cluster

Parameters

- **dives** – a Pandas DataFrame of dive profiles to export
- **data** – a Pandas dataframe of the original dive data
- **folder** – a string indicating the parent folder for the files and sub folders
- **is_surface_events** – a boolean indicating if the dive profiles are entirely surface events

divebomb.**export_to_csv**(*folder*, *dives*, *loadings*, *pca_output_matrix*, *insufficient_dives=None*)

Will output dive profiles, loadings, PCA Matrix, and inssufficent dive into the indicated folder as CSVs.

Parameters

- **folder** – the path to export all files to, the folder will be overwritten
- **dives** – a Pandas DataFrame of the dive profiles and clusters, usually generated from `cluster_dives()`
- **loadings** – a Pandas DataFrame of the Principle Component Analysis loadings from `cluster_dives()`
- **pca_output_matrix** – a Pandas DataFrame of the Principle Component Analysis results from `cluster_dives()`
- **insufficent_dives** – a Pandas DataFrame of dives that could not be profiled from `cluster_dives()`

divebomb.**export_to_netcdf**(*folder*, *data*, *dives*, *loadings*, *pca_output_matrix*, *insufficient_dives=None*)

Will output dive profiles, loadings, PCA Matrix, and inssufficent dive into the indicated folder as netCDF files. Additionally subfolders will be output by cluster with separate files for each dive.

Parameters

- **folder** – the path to export all files to, the folder will be overwritten
- **dives** – a Pandas DataFrame of the dive profiles and clusters, usually generated from `cluster_dives()`
- **loadings** – a Pandas DataFrame of the Principle Component Analysis loadings from `cluster_dives()`
- **pca_output_matrix** – a Pandas DataFrame of the Principle Component Analysis results from `cluster_dives()`
- **insufficent_dives** – a Pandas DataFrame of dives that could not be profiled from `cluster_dives()`

divebomb.**get_dive_starting_points**(*data*, *dive_detection_sensitivity*, *is_surfacing_animal=True*, *minimal_time_between_dives=120*, *surface_threshold=0*, *columns={'depth': 'depth', 'time': 'time'}*)

Parameters

- **data** – a dataframe needing a time and a depth column
- **is_surfacing_animal** – a boolean indicating whether it's an animal that is gaurantedd to surface between dives
- **dive_detection_sensitivity** – a value bteween 0 and 1 indicating the peak detection threshold, the lower the value the deeper the threshold
- **minimal_time_between_dives** – the minimum time in seconds that needs to occur before there can be a new dive segement
- **surface_threshold** – the threshold at which is considered surface for surfacing animals, default is 0
- **columns** – column renaming dictionary if needed

divebomb.**profile_cluster_export**(*data*, *folder=None*, *columns={'depth': 'depth', 'time': 'time'}*, *is_surfacing_animal=True*, *dive_detection_sensitivity=None*, *minimal_time_between_dives=120*, *surface_threshold=0*, *at_depth_threshold=0.15*)

Calls *profile_dives*, *cluster_dives*, and *export_to_netcdf*

Parameters

- **data** – a dataframe needing a time and a depth column
- **folder** – a parent folder to write out to
- **columns** – column renaming dictionary if needed
- **is_surfacing_animal** – a boolean indicating whether it's an animal that is gauranteed to surface between dives
- **dive_detection_sensitivity** – a value bteween 0 and 1 indicating the peak detection threshold, the lower the value the deeper the threshold
- **minimal_time_between_dives** – the minimum time in seconds that needs to occur before there can be a new dive segement

- **surface_threshold** – the threshold at which is considered surface for surfacing animals, default is 0

Returns two dataframes for the dive profiles and the original data

divebomb.**profile_dives**(*data*, *columns={'depth': 'depth', 'time': 'time'}*, *is_surfacing_animal=True*, *dive_detection_sensitivity=None*, *minimal_time_between_dives=120*, *surface_threshold=0*, *ipython_display_mode=False*, *at_depth_threshold=0.15*)

Calls the other functions to split and profile each dive. This function uses the `divebomb.Dive` or `divebomb.DeepDive` class to profile the dives.

Parameters

- **data** – a dataframe needing a time and a depth column
- **columns** – column renaming dictionary if needed
- **is_surfacing_animal** – a boolean indicating whether it's an animal that is gauranteed to surface between dives
- **dive_detection_sensitivity** – a value bteween 0 and 1 indicating the peak detection threshold, the lower the value the deeper the threshold
- **minimal_time_between_dives** – the minimum time in seconds that needs to occur before there can be a new dive segement
- **surface_threshold** – the threshold at which is considered surface for surfacing animals, default is 0
- **ipython_display_mode** – whether or not to display the dives

Returns two dataframes for the dive profiles, inssufficient dives, and the original data

7.5 Dive Class

The Dive class is used to encapsulate all the attributes of a dive and the data needed to reconstruct a plot of the dive. The `at_depth_threshold` defaults to `0.15` which means anything below `85%` of the depth of the dive is considered to be at depth.

`surface_threshold` is used to determine how shallow the animal should be before it is considered to be at the surface. It defualts tp `0` but can be adjusted if the animal is large or you want a larger depth window for surface behaviours. `surface_threshold` should be passed in meters.

class Dive.**Dive**(*data*, *columns={'depth': 'depth', 'time': 'time'}*, *surface_threshold=0*, *at_depth_threshold=0.15*)

Variables

- **max_depth** – the max depth in the dive
- **dive_start** – the timestamp of the first point in the dive
- **dive_end** – the timestamp of the last point in the dive
- **bottom_start** – the timestamp of the first point in the dive when the animal is at depth
- **td_bottom_duration** – a timedelta object containing the duration of the time the animal is at depth in seconds
- **td_descent_duration** – a timedelta object containing the duration of the time the animal is descending in seconds

- **td_ascent_duration** – a timedelta object containing the duration of the time the animal is ascending in seconds
- **td_surface_duration** – a timedelta object containing the duration of the time the animal is at the surface in seconds
- **bottom_variance** – the variance of the depth while the animal is at the bottom of the dive
- **dive_variance** – the variance of the depth for the entire dive.
- **descent_velocity** – the average velocity of the descent
- **ascent_velocity** – the average velocity of the ascent
- **peaks** – the number of peaks found in the dive profile
- **left_skew** – a boolean of 1 or 0 indicating if the dive is left skewed
- **right_skew** – a boolean of 1 or 0 indicating if the dive is right skewed
- **no_skew** – a boolean of 1 or 0 indicating if the dive is not skewed
- **insufficient_data** – a boolean indicating whether or not the profile could be completed

get_ascent_duration(*at_depth_threshold=0.15*)
This function also sets the bottom duration.

Parameters **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns the ascent duration in seconds

get_ascent_velocity()

Returns the ascent velocity in m/s

get_descent_duration(*at_depth_threshold=0.15*)

Parameters **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns the descent duration in seconds

get_descent_velocity()

Returns the descent velocity in m/s

get_peaks(*surface_threshold=0*)

Returns number of peaks found within a dive

get_surface_duration()

Returns the surface duration in seconds

plot()

Returns a plotly graph showing the phases of the dive

set_bottom_variance()
This function also set total dive variance

Returns the standard variance in depth during the bottom portion of the dive in meters

set_dive_variance()

Returns the standard variancet in depth during dive in meters

set_skew()
Sets the objects skew as left, right, or no skew

to_dict()

Returns a dictionary of the dive profile

7.6 DeepDive Class

The Dive class is used to encapsulate all the attriubtes of a dive from a non-surfacing animal and the data needed to reconstruct a plot of the dive. The `at_depth_threshold` defaults t0 `0.15` which means anything below `85%` of the relative depth of the dive is considered to be at depth.

class DeepDive.**DeepDive**(*data*, *columns={'depth': 'depth', 'time': 'time'}*, *at_depth_threshold=0.15*)

Variables

- **max_depth** – the max depth in the dive
- **min_depth** – the max depth in the dive
- **dive_start** – the timestamp of the first point in the dive
- **dive_end** – the timestamp of the last point in the dive
- **td_total_duration** – a timedelta (in seconds since 1970-01-01) containing the duration of the dive
- **depth_variance** – the variance of the depth for the entire dive.
- **average_vertical_velocity** – the mean velocity of the animal over the entire dive with negative value indicating upward movement
- **average_descent_velocity** – the average velocity of any downward movement as positive value
- **average_ascent_velocity** – the average velocity of any upward movement as positive value
- **number_of_descent_transitions** – the number of times and animal moves descends any distance in the dive period
- **number_of_ascent_transitions** – the number of times and animal moves ascends any distance in the dive period
- **total_descent_distance_traveled** – the total absolute distance in meters in which the anaimal moves down
- **total_ascent_distance_traveled** – the total absolute distance in meters in which the anaimal moves up
- **overall_change_in_depth** – the difference between the minimum and maximum depth within the dive period
- **td_time_at_depth** – the duration in seconds at which the animal spends in the deepest part of the vertical movement (< 85% depth)
- **td_time_pre_depth** – the duration in seconds befor the deepest part of the vertical movement (< 85% depth)
- **td_time_post_depth** – the duration in seconds after the deepest part of the vertical movement (< 85% depth)

- **peaks** – the number of peaks found in the dive profile
- **left_skew** – a boolean of 1 or 0 indicating if the dive is left skewed
- **right_skew** – a boolean of 1 or 0 indicating if the dive is right skewed
- **no_skew** – a boolean of 1 or 0 indicating if the dive is not skewed

get_ascent_vertical_distance()

Returns the total vertical distance travelled downwards in meters

get_average_ascent_velocity()

Returns the average upwards velocity in m/s

get_average_descent_velocity()

Returns the average downwards velocity in m/s

get_descent_vertical_distance()

Returns the total vertical distance travelled upwards in meters

get_peaks()

Returns number of peaks found within a dive

get_time_at_depth(*at_depth_threshold=0.15*)

Parameters **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns the duration at depth in seconds

get_time_post_depth(*at_depth_threshold=0.15*)

Parameters **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns the duration after depth in seconds

get_time_pre_depth(*at_depth_threshold=0.15*)

Parameters **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns the duration before depth in seconds

plot()

Returns a plotly graph showing the phases of the dive

set_skew()
Sets the objects skew as left, right, or no skew

to_dict()

Returns a dictionary of the dive profile

7.7 Preprocessing Functions

The preprocessing module is used help correct dive drift and offsets. The offset is calculated using a rolling time window, similar to what is explained here.

There are two methods for the main function, `correct_depth_offset()`:

- max: zeros the local maxium and uses the difference as the offset for the rest
- mean: uses the time window and a maximum depth to look for the average offset within the window

divebomb.preprocessing.**calculate_window_mean**(*window*, *surface_threshold*, *df*)

Parameters

- **window** – an int to determine the size for a rolling median
- **surface_threshold** – the maximum depth that will be considered for the offset
- **df** – Pandas Dataframe of the dive data

Returns An average offset in meters using the defined window

divebomb.preprocessing.**correct_depth_offset**(*data*, *window=3600*, *columns={'depth': 'depth', 'time': 'time'}*, *aux_file='corrected_depth_auxillary_data.nc'*, *method='max'*, *surface_threshold=4*)

Parameters

- **data** – The dataset consisting of a time and a depth column
- **window** – time window (in seconds) to use in the calculation
- **aux_file** – A netCDF file to write all of the calculated offsets and window size
- **columns** – column renaming dictionary if needed
- **method** – either 'max' or 'mean' declaring the calculation method, default is max
- **surface_threshold** – maximum values (in meters) to use when using the mean the calculate

Returns A DataFrame with a corrected depth

divebomb.preprocessing.**zlib_encoding**(*ds*)

This is a helper function for xarray to compress all variables going to netCDF

Parameters **ds** – an xarray Dataset

Returns A dictionary indicating zlib compression for all variables

7.8 Plotting Functions

The plotting module can be used to plot the dives from the output netCDF files. `plot_from_nc()` will plot a single dive separated into its pahses and `cluster_summary_plot()` will five the minimum, maximum, and average depth at time (seconds) into the dive for each cluster.

divebomb.plotting.**cluster_summary_plot**(*folder*, *ipython_display=True*, *filename='index.html'*, *title='Dive Cluster Summary'*, *scale={'depth': False, 'time': False}*)

Parameters

- **folder** – the path to the results folder contianing the cluster folders
- **ipython_display** – a boolean indicating whether or not to show the dive in a notebook
- **filename** – the filename to save the dive to if it is not shown in a notebook
- **title** – the displaye title of the plot

Returns a plotly graph summary of all of the dive clusters

divebomb.plotting.**plot_deepdive_from_nc**(*folder*, *cluster*, *dive_id*, *ipython_display=True*, *filename='index.html'*, *at_depth_threshold=0.15*)

Parameters

- **folder** – the path to the results folder contianing the cluster folders
- **cluster** – the number of the cluster of the dive
- **dive_id** – the number of of the dive
- **ipython_display** – a boolean indicating whether or not to show the dive in a notebook
- **filename** – the filename to save the dive to if it is not shown in a notebook
- **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns a plotly line chart of the dive

divebomb.plotting.**plot_dive_from_nc**(*folder*, *cluster*, *dive_id*, *ipython_display=True*, *filename='index.html'*, *at_depth_threshold=0.15*, *title='Clusters'*)

Parameters

- **folder** – the path to the results folder contianing the cluster folders
- **cluster** – the number of the cluster of the dive
- **dive_id** – the number of of the dive
- **ipython_display** – a boolean indicating whether or not to show the dive in a notebook
- **filename** – the filename to save the dive to if it is not shown in a notebook
- **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth
- **title** – string title of plot

Returns a plotly line chart of the dive

divebomb.plotting.**plot_from_nc**(*folder*, *cluster*, *dive_id*, *ipython_display=True*, *type='dive'*, *filename='index.html'*, *at_depth_threshold=0.15*)

Parameters

- **folder** – the path to the results folder contianing the cluster folders
- **cluster** – the number of the cluster of the dive
- **dive_id** – the number of of the dive
- **type** – a string of either either `dive` or `deepdive`
- **ipython_display** – a boolean indicating whether or not to show the dive in a notebook
- **filename** – the filename to save the dive to if it is not shown in a notebook
- **at_depth_threshold** – a value from 0 - 1 indicating distance from the bottom of the dive at which the animal is considered to be at depth

Returns a plotly line chart of the dive

Python Module Index

d

Index