

# Coastal Risk Information Service (C-RISe)

## Python Code Guidelines

### Introduction

This note provides guidelines on the use of the python routines provided with the C-RISe data

#### 1) Python environment

- Install Anaconda python 3 [www.anaconda.com/products/individual](http://www.anaconda.com/products/individual)
- Open Anaconda-Navigator application
- Select “Environments”
- Select “Create” and give a name, e.g. c-rise2020
- On the right is the installed package listing
- Select “Not installed” at the top
- In the “Search Packages” box enter “numpy”
- Check the box to the left of “numpy” and select “Apply”
- Select “Apply” in the “Install Packages” window
- Repeat the last 3 steps for “netcdf4”, “matplotlib”, “scipy”, “basemap”

#### 2) Using Spyder to open, edit and run python code

- Select “Home”
- Ensure the subtitle is “Applications on c-rise2020” (or your chosen environment name)
- Select “Install” under the “Spyder” application
- Once installed select “Launch” under the “Spyder” application
- Open a python program, e.g. windstats\_point\_py3.py
- The code appears on the left of the Spyder window in an editor
- The first statements after the code head import the required python packages
- Under this is the location of the data directory:
- `datdir = '/Users/ellis/Data/L4winds'`
- Change this to the data directory on your computer
- Use the green arrow at the top (7<sup>th</sup> icon along) to save and run the code
- The code will run in the interactive python area bottom right
- To see the output select “Plots” in the area to the top right
- You can also inspect the saved plot

#### 3) Analysis of Wind, Wave Climatology Data

##### Directory Structure

- The Wind Wave code is at:  
C:\Shared\Software\Windows\WindWave
- The code requires the wind and wave climatology data in /Shared/Data/WindWave/winds, and /Shared/Data/WindWave/waves
- Before running them check that the routines have the correct directory path to the source data files. If they have not then they should be edited so that they have the correct paths (e.g. in spyder or a text editor such as notepad)
- Some routines will save time series and plots. They will save these to /Shared/Results/.
- You may need to change this path, again it can be edited in spyder or on a text editor such as notepad.

### Overview of Routines

- To run all the following routines,
  - Open Anaconda-Navigator application
  - Select “Environments”
  - Choose c-rise2020 and “Open Terminal”
  - Change directory to the location of the windwave code (e.g. C:\Shared\Software\Windows\WindWave ), and type python
  - and from inside python window type

```
>>> exec(open("programname.py").read())  
..and answer the prompts
```

- *wavestats\_CCI\_point\_py3.py* – Generates graphs of monthly mean wave heights and distribution at a location – saves figure to /Shared/Results/.
- *wavestats\_CCI\_point\_timeseries\_py3.py* – Generates time series of monthly mean wave heights and distribution at a location – saves figure to /Shared/Results/.
- *wavestats\_CCI\_region\_py3.py* – Generates a map of monthly mean wave heights (python 3.5).
- *windstats\_point\_py3.py* – Generates a graph of monthly mean wind speed at a location – saves figure to /Shared/Results/.
- *windstats\_point\_timeseries\_py3.py* – Generates a time series of monthly mean wind speed at a location – saves data and figures to /Shared/Results/.
- *windstats\_region\_py3.py* – Generates a map of mean monthly wind speed (and direction if the region is small enough - 4° x 4° or less) for a selected area.
- *windstats\_region\_seasonal\_py3.py* - Generates a map of mean 3- monthly wind speed (and direction if the region is small enough - 4° x 4° or less) for a selected area.  
To save figures click on the disk icon

## 4) Analysis of Surface Current Data

### Directory Structure

- The Surface Current code is at:  
C:\Shared\Software\Windows\Currents
- The code requires the Surface Current climatology data in  
/Shared/Data/Currents/currents\_L4
- Before running them check that the routines have the correct directory path to the source data files. If they have not then they should be edited so that they have the correct paths (e.g. in spyder or a text editor such as notepad)
- this path, again it can be edited in spyder or on a text editor such as notepad.
- We hope to provide updated surface current data later, and will provide new python code to analyse this data set.

### Overview of Routines

- To run all the following routines,
  - Open Anaconda-Navigator application
  - Select “Environments”
  - Choose c-rise2020 and “Open Terminal”
  - Change directory to the location of the Currents code (e.g. C:\Shared\Software\Windows\Currents ), and type python
  - and from inside python window type

```
>>> exec(open("programname.py").read())  
..and answer the prompts
```

- *curplot\_point\_month.py* – Generates a time series of current speed and direction for the selected year, and also for selected months. You will be prompted for new months, until you enter return. To save the files, click on the disc icon.
- *currents\_region\_day.py* – Generates a plot of daily mean current speed (and direction if the region is small enough - 4° x 4° or less) for the selected region, year, month and day. To save the file, click on the disc icon.
- *currents\_region\_month.py* – Generates a plot of monthly mean current speed (and direction if the region is small enough - 4° x 4° or less) and standard deviation for the selected region, year and month. To save the files, click on the disc icon.
- *currents\_region\_monthly.py* – Generates a plot of monthly mean (across all years) current speed (and direction if the region is small enough - 4° x 4° or less) and standard deviation for the selected region and month. To save the files, click on the disc icon.

## 5) Analysis of Sea Level Data

### Directory Structure

- The Sea Level code is at:  
C:\Shared\Software\Windows\SeaLevel\SeaLevelValidation
- Folder python\_code/ contains the code for the validation
- All figures generated by the code are saved to figures/
- Statistics and processed data are saved to data/
- The code requires the along-track time series altimeter data (<mission>\_stats\_ales\_P<cycle>.nc) in /Shared/Data/C-RiSe/AltimetryByPass

### Performing Validation at a Selected Tide Gauge Station

- Open Anaconda-Navigator application
- Select “Environments”
- Choose c-rise2020 and “Open Terminal”
- Change directory to the location of the sea level code (e.g. C:\Shared\Software\Windows\SeaLevel\SeaLevelValidation\python\_code )
- Next, suppose that you want to validate data at Pemba using Envisat/Altika data (the second parameter of the function is the altimetry dataset, which can take the values ‘j1j2j3’ (Jason missions) and ‘n1sa’ (Envisat/Altika)):

```
$ python genStats.py Pemba n1sa
```

- The above command will produce figures showing the correlation and root mean square (RMS) difference as a function of distance from the coast for the closest altimetry passes. It will also produce a figure comparing the sea level time series from altimetry and the tide gauge for the “distance from coast” providing the maximum correlation. Finally, it will also generate a map showing the location of the tide gauges and the selected altimetry passes. All figures will be saved to folder figures/.
- Note that the performance of the altimetry data is assessed as a function of distance from the coast by assigning such data to distance bands of X km width starting from the 0-X km band. Only data that fall within Y km of the tide gauge are used. For each altimetry pass we obtain one altimetry value by averaging all the records falling within the selected distance band. Records with an absolute sea level anomaly larger than 2 m or three standard deviations above the mean are rejected prior to computing the average. The corresponding tide gauge matching value is obtained by linearly interpolating the tide gauge observations to the time of the altimetry pass.
- The width of the distance bands, X, and the maximum distance from the tide gauge, Y, can be adjusted by editing lines 12 and 13 of genStats.py (python\_code/).

- Note: you can run `genStats.py` also from within the Python interpreter. To do that, start the Python interpreter from the shell and run the following commands:

```
>>> from genStats import genStats
>>> genStats ("Pemba", "n1sa")
```

### Performing Validation at all tide gauges

- Open Anaconda-Navigator application
- Select "Environments"
- Choose `c-rise2020` and "Open Terminal"
- Change directory to the location of the sea level code (e.g. `C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code` )

```
$ python genStatsallTg.py n1sa
```

- This function takes one parameter denoting the altimetry dataset: 'j1j2j3' (Jason missions) and 'n1sa' (Envisat/Altika). Note that this will take some time to complete.
- The width of the distance bands,  $X$ , and the maximum distance from the tide gauge,  $Y$ , can also be adjusted by editing lines 11 and 12 of `genStats_allTg.py` (`python_code/`).

### Plotting sea-level maps

- Open Anaconda-Navigator application
- Select "Environments"
- Choose `c-rise2020` and "Open Terminal"
- Change directory to the location of the sea level code (e.g. `C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code` )
- The plotting routine needs the input file: `"\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\data\statsAltiTracks_mission"`. Where "mission" is `n1sa` or `J1j2j3`.
- If this file is missing, then `"plotStatsMap.py"` will not run. This file is created by running `"genStatsTracks.py"` as follows (n.b. It can take a long time (~8 hrs) to run):

```
$ python genStatsTracks.py mission ("mission" is n1sa or J1j2j3)
```

- Function `plotStatsMap.py` allows us to plot maps of a preselected domain showing the amplitude and phase of the sea-level annual and semi-annual cycles, the spatial distribution of sea-level trends, and the standard deviation of detrended and deseasoned sea-level anomalies. Type:

```
$ python plotStatsMap.py lon_min lon_max lat_min lat_max stat mission
```

- where `lon_min`, `lon_max`, `lat_min`, `lat_max` denote the four vertices of a rectangle encompassing the domain you are interested in. They represent coordinates and have units of degree. The fifth input "stat" denotes the statistic you want to plot:
- `amp_an` : amplitude of the annual cycle
- `phase_an` : phase of the annual cycle

- `amp_semian` : amplitude of the semi-annual cycle
- `phase_semian` : phase of the semi-annual cycle
- `trend` : trend
- `sdev` : standard deviation of detrended and deseasoned sea-level anomalies
- The last input 'mission' identifies the altimetry dataset: 'j1j2j3' (Jason missions) and 'n1sa' (Envisat/Altika).
- As an example, the following will produce a map of regional trends in the domain (10°E-65°E,40°S-10°S):

```
$ python plotStatsMap.py 10 65 -40 -10 trend n1sa
```

### Plotting time series of sea-level anomalies for selected points on map

- Open Anaconda-Navigator application
- Select "Environments"
- Choose c-rise2020 and "Open Terminal"
- Change directory to the location of the sea level code (e.g. C:\Shared\Software\Windows\SeaLevel\SeaLevelValidation\python\_code )
- The function `plotSeriesSelect.py` allows us to select a point along a satellite track and plot the sea-level anomalies for that point.
- To run it, start the "ipython" interpreter, and :

```
$ ipython  
>>> from plotSeriesSelect import plotSeriesSelect  
>>> plotSeriesSelect('n1sa')
```

- This function takes one parameter denoting the altimetry dataset: 'j1j2j3' (Jason missions) and 'n1sa' (Envisat/Altika). The statement above will open a map showing all the altimetry tracks. Select a point on the tracks using the mouse. This will produce a figure of the sea-level anomalies at the selected point (it might take several seconds while it extracts the data), and also a figure of the tracks with the selected location highlighted. Save the figures by clicking on the file icon.
- To exit the iPython interpreter  

```
>>> quit()
```
- The sea level anomaly time series, and times (in decimal years), will be saved in separate files created under `.\data`. They can be imported to excel and combined to generate a time series for further analysis.