Climate Modelling Package

User Guide

Produced by: Adanna Akwataghibe

August 14, 2019

If you come across any problems, see section ?? for possible solutions or contact me at aa14415@ic.ac.uk

Table of Contents

SECTION 1

Getting started with the Climate Modelling Package

The Climate Modelling package makes use of many python modules and libraries. These libraries must be installed in order to successfully run the program. Please follow the steps the download the package and install the required dependencies.

1.1 Downloading package

The package is stored in GitHub, a repository hosting service. You can either download the folder manually or through the Linux terminal.

1.1.1 Manual download

1. Go to the the repository linked here.
2. Go the upper right hand corner and click on the **Clone or download** button. Then click on **Download Zip**.
3. Extract the folder from the zipped file that has downloaded.
4. Go to a command line terminal and go to the directory where the folder is stored. Then:

```
$ cd
    acse-9-independent-research-project-AdannaAkwats-master
```

5. To check the contents in the folder, do:

```
$ ls
```

If you do not see the folder **ClimateModelling** in the list, then you may need to **cd** again.

If you can see the folders **ClimateModelling** and **ClimateModelling_Parallel**, then the package has successfully downloaded.

1.1.2 Terminal download (Suitable for Linux and Mac)

1. Go to the terminal and check that since we are installing through *Git*, we have to ensure that it is installed. Type on the command line:

```
$ git --version
```

If this gives an output like `git version 2.17.1` (the numbers can be different), then git is already installed.

If git is not installed, then to install type on the command line:

```
$ sudo apt install git-all
```

After it has finished installing, check for the version (as in the previous step) to make sure it has installed properly.

2. To download from git, you have to clone the repository:

```
$ git clone https://github.
  com/msc-acse/acse-9-independent-research-project-AdannaAkwats
  .git
```

3. Go into the folder to make sure it is cloned properly

```
$ cd acse-9-independent-research-project-AdannaAkwats
```

and

```
$ ls
```

If you can see the folders **ClimateModelling** and **ClimateModelling_Parallel**, then the package has successfully downloaded.

1.2 Installing all requirements

1. First ensure that Python is installed on the machine.

```
$ python --version
```

If you can see a version number, then Python is installed.

If Python is not installed, then install it by:

```
$ sudo apt-get update
$ sudo apt-get install python3.6
```

2. You need to install Miniconda in order to get some of the libraries.

```
$ wget https://repo.continuum.
    io/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
$ chmod +x Miniconda3-latest-Linux-x86_64.sh
```

```
$ ./Miniconda-latest-Linux-x86_64.sh
```

   and follow the commands. Then after the installation, restart the terminal so that all the changes can take effect.

   After restarting, check conda has installed by checking the version:

```
$ conda --version
```

   We can now install the rest:

```
$ conda install -c conda-forge iris
```

   And for the plotting:

```
$ conda install basemap
```

3. To install the rest of the dependencies

```
$ pip install -r INSTALL.txt
```

4. If pip is not installed, then install pip with:

```
$ sudo apt-get python-pip
```

   and then go back to the previous step and install the rest of the dependencies.

We now have all packages installed!
Some example code is available in section 3 to run.

SECTION 2

Package structure

The package is structured as follows.

```
    acse-9-independent-research-project-AdannaAkwats/
|-- ClimateModelling_Parallel/
|   |-- ... # Parallelised version of ClimateModelling/
|
|-- ClimateModelling/
|   |-- DATA/
|   |-- INPUT/
|   |-- RESULTS/
|       |-- ensemble_averages/
|
|   |-- directories.py
|   |-- message.log
|   |-- main.py
|
|-- INSTALL.txt
|-- README
```

- `DATA/` : The default folder where NetCDF model output files are stored. You can redirect this directory in `directories.py` if the data is stored somewhere else.
- `INPUT/` : Folder where all input files (e.g `input.py`, mask files etc) are stored.

- `RESULTS/` : Folder where all results and files from software run is stored. In `ensemble_averages`, the netCDF output files from program are stored.
- `message.log` : File where all progress and error messages are stored.

SECTION 3

Running the software

Enter the **ClimateModelling** folder:

```
$ cd ClimateModelling
```

There are 2 ways to run the software, either using an input file or using command line arguments.

1. To run the `input_example.txt` file which calculates the mean of air temperature in North America and plots the map and histogram:

   ```
   $ python main.py -ex
   ```

   After running, it should print a SUCCESS message on the command line and show the plots.

   NOTE: If you run this on a terminal does not allow plotting, an error will be thrown. So you have to set plot to "False" in the input file.

2. After filling the arguments in `input.txt`, the code can be run with this input by:

   ```
   $ python main.py
   ```

   The files `input.txt` and `input_example.txt` also contain in-depth information on what each argument means and its expected parameters.

3. You can also use the command line, e.g. to run with the `input_example.txt` arguments:

   ```
   $ python main.py E1_north_america 1970 -v air_temperature
     -e 1 -a mean -p 1 -o
   ```

For help on what each argument means:

```
$ python main.py --help
```

4. After each run, you can check the progress or potential error messages by:

```
$ less message.log
```

4 SECTION

Some applications

The following are some scenarios that you may want to run.

4.0.1 Calculate mean sea surface temperature between 1980 and 1990 at grid point (14.5, -27.5) and output a time series, histogram and NetCDF file with mean data

```
# REQUIRED ARGUMENTS
# -----------------------------------------------------
Prefix: sst
Start date of analysis: 1980
Variables: sst
Number of ensembles: 1
#
# -----------------------------------------------------
# OPTIONAL ARGUMENTS
# -----------------------------------------------------
End date of analysis: 1990
Analysis: mean
Total ensemble stats: False
Plot: 1
Monthly: False
Grid: 14.5, -27.5
Sample:
Mask file:
```

```
Save Output: True
Covary: False
Histogram bin selection: fd
Longitude centre:
Save extract data: False
User function:
Calculate areas: False
```

4.0.2 Calculate mean sea surface temperature between 1980 and 1990 using region of NINO-3 and output histogram, map and NetCDF file with mean data

```
# REQUIRED ARGUMENTS
# -------------------------------------------------
Prefix: sst
Start date of analysis: 1980
Variables: sst
Number of ensembles: 1
#
# -------------------------------------------------
# OPTIONAL ARGUMENTS
# -------------------------------------------------
End date of analysis: 1990
Analysis: mean
Total ensemble stats: False
Plot: 1
Monthly: False
Grid:
Sample:
Mask file: nino3_mask.txt
Save Output: True
Covary: False
Histogram bin selection: fd
Longitude centre:
Save extract data: False
User function:
Calculate areas: False
```

4.0.3 Calculate mean and std of dic_deltap between 2005 and 2010, with longitude centred at -30 degrees east and also grid boxes areas calculated. Outputs are NetCDF files containing areas and mean and std.

```
# REQUIRED ARGUMENTS
# -------------------------------------------------
Prefix: dic_deltap
```

```
Start date of analysis: 2005
Variables: dic_deltap
Number of ensembles: 1
#
# ----------------------------------------------------
# OPTIONAL ARGUMENTS
# ----------------------------------------------------
End date of analysis: 2010
Analysis: mean, std
Total ensemble stats: False
Plot: False
Monthly: False
Grid:
Sample:
Mask file:
Save Output: False
Covary: False
Histogram bin selection: fd
Longitude centre: -30
Save extract data: False
User function:
Calculate areas: True
```