# Guide to the Python Code

This guide has been prepared to accompany the Honours thesis *Observed and Simulated Relationships Amongst ENSO, the IPO, and Rainfall Variability in Eastern Australia*, written by Sonya J. Wellby and submitted to the Fenner School of Environment and Society of the Australian National University in November 2015. This guide provides an overview of each of the Python files created for the analysis (available in the electronic appendix folder "Code/Python code", or on GitHub (https://github.com/sonyawellby/anu_honours)). The files are presented in the same order as they are called upon in the analysis.

## 1. Preparation for analysis:

These files set the working directory, call the observed and modelled datasets and prepare these for further analysis, and to establish parameters for use in later functions.

### *cwd.py*

This file sets the working directory.

### *foo.py*

This file prints dummy code if the function cwd() (in cwd.py) is called and the working directory has been set properly.

### *data.py*

This file specifies the filepaths of the eight NetCDF datasets used in this analysis: six ACCESS1.3 datasets, one HadISST1.1 dataset, and two versions of the AWAP dataset (interpolated and uninterpolated). This filepaths can be edited to reflect the location of the NetCDF datasets on your local computer.

This file also provides URLs for where the NetCDF datasets can be downloaded from.

### *parameters.py*

This file defines parameters that are called on by functions in other files. As these parameters are often drawn on by multiple files, it is important that they be defined in this "central" file to avoid error. Parameters defined here include:

- The base period (1961–1990); other base periods used in the base period comparison are also defined here.
- Parameters for the Chebyshev low-pass filter used in calculating the filtered TPI.
- The number of standard deviations that define an ENSO or IPO state. This is either set to 0.5 (if analysing the TPI according to Henley *et al.* (2015)), 2.0 or 3.0 (if analysing

oscillation states defined as two or three standard deviations above or below the index means).

### *access_prepare_pr.py, awap_prepare.py, access_prepare_ts.py, hadisst_prepare_pr.py*

These files prepare original data (AWAP, ACCESS1.3 precipitation, ACCESS1.3 SST, HadISST1.1) for further analysis.

- In the case of ACCESS1.3 precipitation and ACCESS1.3 SST data, asks for user input on which files are being analysed.
- Latitude and longitudes are defined within the ACCESS1.3 and HadISST1.1 NetCDF files but not in the AWAP NetCDF file. The awap_prepare.py file assigns latitudes and longitudes to the AWAP dataset.
- Necessary conversions are made so that inter-dataset comparisons can be made. AWAP data is converted from metres/day to mm/day. ACCESS1.3 precipitation data is converted from $kg/m^2$/second to $kg/m^2$/day. HadISST1.1 data SSTs are converted from Kelvin to degrees Celsius.
- Invalid values are masked. AWAP and ACCESS1.3 precipitation datasets are masked for values below zero. HadISST1.1 data is masked for values less than –10ºC, and values at the dateline are masked to account for a discontinuity in the dataset at the dateline after 1982 (http://www.metoffice.gov.uk/hadobs/hadisst/). ACCESS1.3 SST data is masked for values less than –2ºC and greater than 35ºC, as this dataset provides SSTs over the oceans and "skin surface temperature" over land (temperatures that are not realistic for SSTs have been masked).
- Data is trimmed to the time period of interest (June 1900–May 2005) and broken into annual, seasonal, and monthly components for further analysis. An extended version of these datasets are made available for the HadISST1.1 and ACCESS1.3 SST (January 1900–December 2005) to allow for the running mean to be computed for Niño 3.4. An extended version of the AWAP dataset is also made available so that interpolated data can be mapped for the same time period as uninterpolated precipitation data (which is only available in the annual format); this is necessary for Appendix 1.
- Data relating to latitude, longitude, and units are made available for use in other scripts.

### *access_trim.py*

This file takes the output from access_prepare_pr.py and reduces the grid to the same coordinates as AWAP. This data is made available for further analysis.

## 2. Index generation:

These files generate the Niño 3.4 and TPI indices.

### *enso.py, tpi.py*

These files contain the functions called upon in enso_csv.py and tpi_csv.py. The Niño 3.4 and TPI regions in the Pacific Ocean are defined, and functions are presented to divide oscillation datasets according to their positive, neutral and negative states (both according to Trenberth (1997) and Henley *et al.* (2015), as well as ±2σ and ±3σ above the index means).

### *enso_csv.py*

This file produces monthly Niño 3.4 values for the study period (June 1900–May 2005), which is a total of 1260 months. It makes this data available both in NetCDF and CSV format (currently, the CSV file is created in the "data/" folder, but, if desired, this can be manually changed to a preferred location on your own computer). The Niño 3.4 dataset is divided into positive, neutral, and negative ENSO states for annual and seasonal data. (Monthly ENSO states are defined in indices_phase.py.)

The Niño 3.4 index is formed from the hadisst_prepare.py and access_prepare_ts.py "extension" datasets (January 1900–December 2005). The following computations are performed.

1. Defines the area that is the Niño 3.4 box in the Pacific Ocean. The dataset is trimmed down to this area.
2. Twelve datasets of the Niño 3.4 area are created (one for each month).
    a. For each month in the base period (1961–1990), the average SST value of each grid cell in the Niño 3.4 area is computed.
3. For each month, the base-period average (*Step 2a*) for each grid cell is subtracted from the SST value of each grid cell, for each time-step, in the overall monthly dataset (*Step 2b*). These are the monthly SST anomalies.
4. The average SST anomaly for the whole Niño 3.4 area is computed for each time-step (i.e. the Niño 3.4 value for Time 1 is the average of all grid-cell SST anomalies in the Niño 3.4 region in Time 1).
5. A five month running-mean is computed for each time-step (March 1900–October 2005), using the monthly mean anomaly data (*Step 4*). This dataset is then cropped to the period of interest (June 1900–May 2005) and broken into pieces of interest (e.g. annual, seasons). Masked copies of the whole dataset are created where only positive, neutral or negative ENSO years are visible. Positive, neutral, and negative years are defined either according to (a) Trenberth et al. (1997), or (b) standard deviations above the Niño 3.4 mean.
6. The entire monthly datasets (unmasked, and the three masked versions) as well as masked and non-masked seasonal and annual datasets are made available for use in other scripts.
7. The entire datasets (Jun 1900–May 2005) for observations and ACCESS1.3 runs are saved to CSV.

### *tpi_csv.py*

This file produces monthly TPI values for the study period (June 1900–May 2005), which is a total of 1260 months (indices are formed from the monthly output of hadisst_prepare.py and access_prepare_ts.py, rather than the entire dataset, to avoid seasonalisation of the data). It makes this data available both in NetCDF and CSV format (currently, the CSV file is created in the "data/" folder, but, if desired, this can be manually changed to a preferred location on your own computer). The TPI dataset is divided into positive, neutral, and negative IPO states for annual and seasonal data. (Monthly IPO states are defined in indices_phase.py.)

1. For each month (twelve subset datasets of the 1900–2005 dataset), the areas that are the three TPI boxes are defined. The monthly dataset is trimmed down to these areas.
2. For each monthly dataset, the base period dataset is created (1961–1990).
    a. The average SST value of each grid cell in the TPI areas is computed for the base period (1961–1990).

3. For each month, the base-period average (*Step 2a*) for each grid cell is subtracted from the SST value of each grid cell, for each time-step, in the overall monthly dataset (*Step 2b*). These are the monthly SST anomalies.
4. The average SST anomaly for the each of the TPI areas is computed for each time-step (i.e. the TPI value for Time 1 in TPI Area 1 is the average of all grid-cell SST anomalies in TPI Area 1 in Time 1).
5. The unfiltered TPI is calculated from this output according to the methods of Henley *et al.* (2015). The filtered TPI is then calculated from the unfiltered TPI by applying a low-pass Chebyshev filter.
6. The filtered TPI data is divided into seasonal and annual data.
7. The entire monthly datasets (unmasked, and the three masked versions—where TPI states are defined according to ±0.5σ, ±2.0σ, and ±3.0σ above the TPI mean) as well as masked and non-masked seasonal and annual datasets are made available for use in other scripts.
8. The entire unfiltered and filtered datasets (June 1900–May 2005) for observations and ACCESS1.3 runs are saved to CSV.

### *indices_phase.py*

This file serves two purposes:

1. It makes monthly subsets of the Niño 3.4 and TPI indices available for analysis. This includes all monthly data *and* monthly data that has been stratified according to ENSO and IPO states (i.e. positive, neutral, negative).
2. It "collects" all Niño 3.4 and TPI data so far generated (i.e. in enso_csv.py and tpi_csv.py) and makes this conveniently available in the one file (so that multiple files do not need to be called).

### *csv2array.py*

This file converts ENSO and IPO indices produced by others (e.g. Mantua *et al.* (1997)) into one dimensional NumPy arrays (i.e. vectors with length 1260). Data needs to be in a compatible format to analyse how well others' indices and those generated in this study are correlated, etc.

## 3. Files to plot maps:

These files contain the main functions used to generate the maps in this analysis. Other scripts commonly call upon the functions defined within these files.

### *maps_sub.py*

This file defines the base-map underlying all maps. It defines the Australian area, outlines (e.g. continent vs. states), grid dimensions, axis labels, and a function to save the images generated.

### *plot.py*

This plot defines the various dictionaries (which contain latitudinal and longitudinal data, units, etc.) called upon in creating more complex maps. If colour-bars are used, their range is defined here. Functions to plot single images (plot()) and multiple images as one image (multi()), as well as several other variants, are defined. A short script is included at the end of the file to generate explanatory figures (e.g. the Köppen climate zone areas, etc.).

### *plot_awap_uninterpolated.py*

This script generates a map of mean observed, uninterpolated precipitation (i.e. AWAP data with the resolution of $0.05^{o} \times 0.05^{o}$).

## 4. Correlation analysis:

These files were used to perform (a) analyses before performing the main correlation analysis, (b) analyses on rainfall–Niño 3.4 and rainfall–TPI correlations, (c) analyses on TPI–Niño 3.4 correlations, and (d) comparisons between the results of various correlation analyses already conducted.

### *time_series.py*

This file generates graphs of precipitation, Niño 3.4 values, and TPI values for the study period (June 1900 to May 2005). Four sets of time series graphs are produced (×1 AWAP; ×3 ACCESS1.3), for monthly, seasonal, and annual data.

### *scatter.py*

This file produces scatterplots of Niño 3.4 values (x-axis) and TPI values (y-axis). Each scatterplot contains 105 points (one for each year in the study period). Four sets of time series graphs are produced (×1 AWAP; ×3 ACCESS1.3), for monthly, seasonal, and annual data.

### *cross_corr.py*

This file defines functions that are called on by cross_corr_routine.py when plotting cross correlations between (a) rainfall and indices, and (b) Niño 3.4 and the TPI.

### *cross_corr_routine.py*

This file produces cross-correlation plots of (a) rainfall and indices, and (b) Niño 3.4 and the TPI. Rainfall–index cross-correlations are performed on (a) monthly (i.e. 1260 time-steps), (b) seasonal (i.e. 420 time-steps), and (c) annual (i.e. 105 time-steps) precipitation data.

### *correlation.py*

This file defines functions that are called on by other scripts in the correlation analysis. Functions are defined that: create vectors of average precipitation (with length 105, for the 105 years in the study), so that precipitation and oscillation indices can be correlated; return rainfall–index correlation arrays for the Australian region so that they can be mapped; and plot correlation maps, including for correlations stratified by oscillation state, and for maps which show differences in correlations between datasets

### *correlation_routine_awap.py, correlation_routine_R1.py, correlation_routine_R2.py, correlation_routine_R3.py*

These scripts perform the following roles:

1. Average correlations (for the 1900–2005 study period) between Niño 3.4 and rainfall, and the TPI and rainfall, are mapped for each grid-cell across Australia. Grid cells where correlations are not statistically significant are masked (i.e. appear white).
2. Single correlation coefficients are produced for indices and rainfall in (a) Australia, (b) eastern Australia, and (c) Köppen climate zones in eastern Australia. A one-dimensional vector of length 105 (for the 105 years in the dataset) is firstly produced containing a mean precipitation value for each year; this is then correlated with a one-dimensional index vector (also of length 105). This output is made available in CSV spreadsheets. If a correlation coefficient is not statistically significant, it appear in the spreadsheet as a "nan" (i.e. invalid) value.
3. Correlation arrays for the whole of Australia are compared between models and observations. For grid cells where both observed and modelled data contain values, the modelled array is subtracted from the observed array, and the difference in correlation is plotted. Where this difference is not significant, it is masked. If the number of grid cells shared by observed and modelled datasets is greater than 30, the statistical significance is determined by a z-statistic; if it is less than 30, a two-tailed t-statistic is used.

### *unpaired_t_test.py*

This file defines two functions for use in other files.

1. **normality**: this function tests a dataset to determine if it is normally distributed or otherwise. It returns the p-value associated with the null hypothesis that the data is normally distributed (i.e. if $p \leq 0.05$ it is not normally distributed).
2. **unpaired_t_test:** this function determines whether or not the means of two independent samples are equal or unequal (null hypothesis: the means of the two samples are equal). If one of the two samples tested are not normally distributed, Levene's test is used to compute the equality of variances between the two samples; otherwise, the equality of variances is tested with an F-test.

### *ttest_correlations.py*

This file tests whether or not the means of arrays of correlation coefficients (between rainfall and indices—as produced in the correlation_routine_xyz.py files described above) are equal, and returns the associated p-values in CSV files. Three types of difference are tested for: (1) significant difference in rainfall–index correlation coefficients generated in Australia and eastern Australia, and eastern Australia and the Köppen climate zones; (2) significant difference between modelled and observed rainfall–index correlation coefficients; and (3) significant difference between modelled rainfall–index correlation coefficients.

### *ENSO_IPO_corr.py*

This file has three main purposes:

1. This script calculates the correlation coefficients between the Niño 3.4 and TPI indices; this output is saved in the CSV file format. The significant difference between modelled Niño 3.4–TPI correlation coefficients is tested (using unpaired_t_test() in unpaired_t_test.py), and the associated p-values are saved in the CSV file format.
2. The normality of the Niño 3.4 and TPI indices (as well as observed and modelled precipitation) is tested, and the associated p-values are saved in the CSV file format.
3. The file defines the function for computing scatterplots of Niño 3.4 and TPI data (used by scatter.py).

### *ENSO_IPO_corr_strat.py*

This script calculates the correlation coefficients between the Niño 3.4 and TPI indices, stratified into the ENSO and IPO states according to (a) the definitions of Trenberth (1997) and Henley *et al.* (2015), and (b) $\pm 2\sigma$ and $\pm 3\sigma$ standard deviations above the index means. The significant difference between Niño 3.4 and TPI stratifications from the ENSO neutral–IPO neutral state is tested (using unpaired_t_test() in unpaired_t_test.py), and the associated p-values are saved in the CSV file format.

## 5. Composite analysis:

These files were used to perform analyses in which precipitation data was stratified in accordance with the positive, neutral, and negative states of the ENSO and IPO.

### *composite.py*

This file defines functions that are called on in scripts performing the composite analysis. This includes functions that mask precipitation data that is not the required ENSO–IPO combination (e.g. if ENSO positive–IPO positive is of interest, precipitation data occurring in the other eight combinations is masked). The functions to plot composite maps are also defined here.

### *difference_maps.py*

This script calculates differences in mean precipitation (mm/day) between the wet and dry states of the two oscillations (i.e. rainfall in ENSO positive minus rainfall in ENSO negative; rainfall in IPO negative minus rainfall in IPO positive). These differences are then plotted for oscillation states defined according to Trenberth (1997) and Henley *et al.* (2015), as well as $\pm 2\sigma$ and $\pm 3\sigma$ above the index means. Maps are produced for observed and modelled rainfall. Where there are no precipitation values in either the positive or negative state of the oscillation in question (i.e. in the ±2σ and ±3σ cases), the entire map is masked (as the difference between the two states cannot be computed).

## *composite_awap.py, composite_R1.py, composite_R2.py, composite_R3.py*

This file produces three-by-three plots of precipitation that has been stratified according to the nine combinations of the positive, neutral, and negative states of ENSO and the IPO. Plots are produced for oscillation states that are defined according to Trenberth (1997) and Henley *et al.* (2015), as well as ±2σ and ±3σ above the index means. Maps are produced for observed and modelled rainfall. Two types of plot are generated: (1) composite maps of mean rainfall (mm/day), and (2) composite maps of mean rainfall anomalies (mm/day), where climatological mean precipitation (i.e. the base period of the month, season or year of interest) is subtracted from mean precipitation stratified according to a particular ENSO–IPO state combination. If no precipitation data exists in the stratification of interest (e.g. ENSO negative–IPO negative combination in the ±3σ case), a masked map is returned (i.e. a blank map).

## *comparison_stratified_rainfall.py*

This file produces three-by-three plots of standardised mean anomalous precipitation that has been stratified according to the nine combinations of the positive, neutral, and negative states of ENSO and the IPO. Plots are produced for oscillation states defined according to Trenberth (1997) and Henley *et al.* (2015), as well as ±2σ and ±3σ above the index means. Maps are produced for observed and modelled rainfall. Standardised mean precipitation anomalies are determined by dividing the precipitation anomaly (described in the section above as '(2)') by the standard deviation of the climatological precipitation (i.e. the base period of the month, season or year of interest). If no precipitation data exists in the stratification of interest (e.g. ENSO negative–IPO negative combination in the ±3σ case), a masked map is returned (i.e. a blank map).

## *stratify_correlate_rainfall_phases.py*

This file produces maps of correlation coefficients between rainfall (that has been stratified according to the states of one oscillation) and the second oscillation. For example, maps are produced of rainfall–TPI correlations, where rainfall has been stratified according to the ENSO states. Only grid cells with statistically significant correlation coefficients are plotted. If no precipitation data exists in the stratification of interest (e.g. ENSO in the ±3σ case), a masked map is returned (i.e. a blank map).