**Hawaii Weather Research Forecast Model Data Processing Instructions**

**Alexander Keyel,** [**akeyel@albany.edu**](mailto:akeyel@albany.edu)

## Getting Started

This workflow requires R. For working with R, I highly recommend using RStudio (https://posit.co/download/rstudio-desktop/).

This guide will cover 2D and 3D data products. The 2D data products were derived from an hourly data set available from the USGS via the internet. The 3D data products are on physical hard-drives at the University of Hawaii Manoa. It will require a person to physically access the hard drives to extract any desired data.

All scripts (and this User’s Manual!) are available on GitHub: [www.github.com/akeyel/HI\_WRF](http://www.github.com/akeyel/HI_WRF).

In file names, ok indicates Oahu/Kauai and hm indicates Hawaii Big Island and Maui. Unfortunately the data conventions for the download differ between the two sets of islands, but once they are converted to daily files, the four islands should all use the same processing steps.

While these scripts were developed for this specific WRF data set, with modification, we anticipate that they could be applied to other WRF data products. The key would be to identify differences in the data format, and apply the minimum number of changes to bring the other data set into the workflow.

## 2D Data

**Accessing the Data**

Tutorial\_DailyDataRDA.R provides a tutorial for using the R format data to perform calculations, and then export the output to a geotif format.

**Repeating the data processing process**

There are three workflows for processing the 2D data sets.

* Workflow\_ppt.R was used to process the precipitation data
  + The precipitation data set required special processing because data were reported in buckets, and it was necessary to process the two bucket-related variables together, and account for some irregularities in the data set (i.e., buckets sometimes exceeded 1).
  + The workflow will need to be run separately for each island.
* Workflow.R was used for non-cumulative variables such as temperature
* Workflow\_Cumulative.R for cumulative variables such as surface runoff.

To get started processing a variable, select the variable of interest, and begin with the appropriate workflow. The Workflow script will contain links to all of the scripts needed to process the data through the different data stages. The example here will use the Workflow.R script as an example. Using the Workflow\_ppt.R and Workflow\_Cumulative.R scripts should be similar. One note is that changes to the code base occurred after the development of Workflow\_ppt.R, so it is possible that some of the helper functions will need to be updated for this script to function properly.

First, update Line 17 of the code to indicate your variable of interest. A complete list of available variables is available in Variables.R. Some variables were missing from the USGS data server data. At least in the case of UDROFF and SFROFF, these are available in the 3D data set. Scripts could be developed to extract and compile this information, but that is outside the scope of the present project, which was to develop scripts to access the USGS data.

For this example, we will use “T2”, the air temperature at 2 meters.

Lines 21 through 32 should be updated to reflect your computer’s paths. This approach was used because the data were in a different directory than the code, due to the size of the data set, and processing was run on two different computers, with different paths on each.

Lines 37 – 39 set up the islands. I typically ran the code interactively, so if I wanted to run it for only one island, I would uncomment line 35, specify the island, and then not run Lines 37 – 40. You need to skip all of the loops around the islands, and just run the interior portions of those code blocks.

Note that there may be some parts that allow you to just run selected blocks (e.g., lines 124-127). This is because some steps may need checking for accuracy, so by setting blocks to 1 will let them run, and setting them to 0 will turn them off. You can also run the blocks interactively in RStudio.

The workflow consists of the following major tasks:

* Load Functions: (Workflow\_hlpr.R)
  + loads functions into memory but does not perform any actions
* Load Settings (e.g., Settings/Settings.R)
  + Runs a settings file, , which contains instructions on how to run the rest of the processing. This will load variables into the active session. This is where you could customize the number of time steps, or the scenarios, for example, or adjust the GMT offset for a new data set.
* Make a grid
  + This creates a grid to facilitate exporting the data to a GIS. For the 2D data, this should not need to be run again, as there is an existing grid for each island, see grids/wrf\_grids, one file for each island ending in \_xy\_grid.csv
* Download the chosen variable(s) for each scenario (05\_Data\_Downloader\_generic.R and 05.5\_CopyData.R).
  + Downloads the data for an island. The speed of this script will depend on your internet connection, and may take several hours per island and scenario. The resulting data download may be large (10’s of GB)
  + Different functions are used for Oahu/Kauai than for Hawaii/Maui, but the script will automatically adjust based on the island name.
* Interpolate missing values
  + This block will add in a missing day (Jan 1, 1996) that was lost due to corruption. It will also add extra hours to the end of the simulation to convert from GMT time to local time (because the data are downloaded in GMT and are being switched to HST).
  + To fill in the missing day for a cumulative variable, the change in amount is divided by 24 and distributed evenly across the 24 hours. To fill in for a non-cumulative variable, it just repeats the immediate previous day. This is only needed for the present day scenario.
  + To add data to the end of the simulation, for the cumulative variables (SRFOFF and UDROFF, the values are just fixed at the prior day’s values for the final hours. For non-cumulative variables where a mean is calculated, the extrapolated values are just set to the recent prior values.
  + Note: this will fill in data in the hourly folder based on the hourly\_raw folder. So if this block is accidentally run twice, it will just overwrite the previously run files, but should not continue to add extra days into the middle of the simulation (it ONLY interpolates that one specific day). You can tell if this script has been run if the hourly folder contains all of the data, ending in 175330 (there will also be the 175320\_deleteme.rda file that contains the interpolated day but not the added extra hours, this can be safely deleted).
* Process the downloaded hourly data into a daily file (07\_ExtractAnnual\_general.R)
  + For the precipitation workflow, Oahu and Kauai process much slower due to the missing I\_RAIN variable and the need to calculate differences hourly
* Provides an option to correct any known errors in the data set. This may vary by variable.
* Create Daily, Annual, and Monthly aggregates from the daily files and create climatologies (10\_ProcessAnnual\_generic.R)
* Convert the annual, monthly, and climatologies to .csv format and geotif format (11\_climatology2geotif.R)
  + These formats should make it easy for GIS users to read in the data.
* Convert the daily files to .csv and geotif format format (12\_Daily2geotif.R)
  + This was not run for everything, because it would create a large number of files and would require a lot of processing time. This can be run individually by users interested in data from specific days.
* Convert the monthly and annual files to GeoTiff (13\_Means2Geotif.R)
* Perform some basic quality control (14\_Quality\_Control\_Write\_up\_Figs.R)
  + If there is a reference file for comparison it may be specified as a reference. Otherwise this will just make descriptive plots of the variables.

## 3D Data

The main output from the 3D data was cloud water interception. To that end, scripts were developed to calculate cloud water interception for selected points, and to estimate it across the entire islands. The calculations are still being finalized and compared to field data, consequently a third version of the script was added to allow someone access with extracted variables to customize the calculations as needed using the pre-selected variables without having access to the full 3D data set.

The cloud water interception required pulling wind data from above the canopy layer. The lowest wind layer above the canopy was used for wind estimation at the canopy level.

The following processing steps were taken to prepare the data set for this analysis:

1. The 3D WRF data set was removed to remove special characters. These characters did not pose a problem on a Linux computer, but made the files unreadable on a Windows machine. This was accomplished using the script XXXXX. This only needs to be done once.
2. Wind heights were extracted (see Wind Height Extraction Process below). This only needed to be done once, so future analyses with this data set can use the extracted data.
3. **Calculate Cloud Water Interception for Selected Points**
   1. The code takes a list of locations provided in .csv format, and converts them to the WRF grid in Lines 52 – 59
   2. Once the location file is pre-processed, simply adjust the island group, scenario, and years, and ensure that the input and output drives are correct, then run the script. It was processing a year in ~5 minutes for Oahu and Kauai and in ~25 - 36 minutes for Hawaii/Maui on my PC.
   3. This requires the full 3D data set
4. **Calculate Cloud Water Interception for an entire island**
   1. 3D\_calculations\_for\_island.R
      1. This is designed to calculate the cloud-water interception for an entire island from the 3D data set (run on the version adjusted to not contain special characters)
   2. 3D\_calculations\_for\_island\_many\_nc\_files.R
      1. This is designed to calculate the cloud-water interception for an entire island from the extracted 3D variables (on an 8 TB hard drive)

**Wind Height Extraction Process**

First, check if the wind height has already been extracted. There should be a folder in the 2D data set hard drive labeled wind\_heights. There should be a sub-folder for each island set and scenario. Hawaii/maui are on the one labeled ‘hawaii’ and Oahu and Kauai are on the one labeled ‘kauai\_oahu’. There should be a year subfolder for each year that has been processed.

* At the time of this documentation writing (2023-12-22), 1999-03-16 was missing for hawaii\_present – this file was corrupted on the 16 TB hard drive

If the wind height has not already been extracted, or you are using this set of code on a different WRF model then:

Open the Extract\_wind\_level\_height.py script with your favorite Python IDE. You will need the netCDF4 module and the wrf module. Note that for installation purposes, the wrf module is the wrf-python module. I had some trouble installing this, but finally got it to work on one computer using the conda install command through my Anaconda installation. (pip install was not going well on a different computer outside of Anaconda).

Here is a link to the documentation for the tool for extracting wind heights:

https://wrf-python.readthedocs.io/en/develop/internal\_api/generated/wrf.g\_geoht.get\_height\_agl.html

Because some of the 16 TB hard drives were very full, I set it up so that the wind heights would be extracted to a different drive than the original data hard drives.

You will need to adjust the paths to fit the input hard drive – each one had a slightly different file structure, but the path\_bits object can be customized to contain the part of the path that changes between years.

It is a good idea to check that the length of the year\_list and either the path\_bits or path\_list match. Use either path\_bits or path\_list. path\_bits if you just want to fill in a small part of each path, use path\_list if you just want to write out the paths you will be using. They are basically two different ways of doing the same thing, so it doesn’t really matter which approach one uses.

NOTE: The year folders CANNOT contain any other files, or the script will crash as written. (exceptions could be added to the script if desired, but was not necessary in this case).

Once the paths are adjusted properly, simply run the script, and it should output each day that is processed. This will give the wind layer heights needed for the cloud water interception wind downscaling calculation.

## Overview of HI\_WRF GitHub Repository and Scripts

**Folders:**

3D: Three scripts for processing the 3D data set into cloud water interception. Note that these require the ten 16 TB hard drives that have the WRF simulation output on it in order to use. The ‘for\_points’ version was written to work with 5 locations with field measurements, while the for\_island version is intended to create island-wide calculations. **NOTE: the calculations have not been finalized yet, so please treat all outputs as preliminary unless further validation/vetting is performed. The 3D\_calculation\_for\_island\_many\_nc\_files.R script allows adjusting from the underlying variables.**

gis/Vector: Public Domain Vector format outlines for each of the Hawaiian Islands downloaded from Natural Earth.

grids/templates: Raster templates from the Hawaii Rainfall Atlas to use to convert the WRF simulation to a standardized raster grid. Note that this involves downscaling lower-resolution data to higher resolution, and many GIS experts recommend going the opposite direction. This was done to take advantage of the standardized grid of the Hawaii Rainfall Atlas and to simplify data comparisons and use.

grids/wrf\_grids: The Lat/Lon points of each WRF model point, along with the row and column indices in the data matrix. These files can be used to look up real-world locations in the WRF simulation, or can be used to geo-locate WRF model outputs by joining them to the grid.

settings/Settings.R and /PrecipSettings.R: Some basic settings for running the Workflow scripts to keep the Workflow scripts tidy. Basically, if you find an undefined object in the Workflow script, chances are it was defined in the settings file. PrecipSettings.R should be used with Workflow\_ppt.R and Settings.R should be used with everything else.

workflows/Workflow.R This is the main script for processing the 2D data. It will call other scripts in the required order. It is best run interactively with one section at a time, as sometimes a step will result in error or may take some time for processing, in which case it may need to be re-run or corrected before attempting the next step. See 2D Data below for details on the scripts used by this workflow.

Workflows/Workflow\_Cumulative.R: A workflow for non-bucket variables where values are given cumulatively over the simulation.

Workflows/Workflow\_ppt.R: A custom workflow for precipitation. This was used to generate the precipitation data files – **but watch for bugs**. The precipitation scripts were later adapted for processing the temperature data, so it is possible that some functions may have changed since they were originally applied to the precipitation data.

wrf\_tables: VEGPARM.TBL was taken from the WRF simulation (NCAR). See WRF\_License.txt for link to WRF and for license terms (public domain).

**Files:**

Variables.R: This script provides an overview of the 2D variables in the WRF file available for use.

Workflow\_hlpr.R This script contains the functions used by the other scripts. It is best navigated by searching for the desired function, as the functions are present in no particular order, and I navigate it using the search option.

## Processing Notes

* January 1 1996 was missing from the present day runs.
  + For rainfall, for Maui and Hawaii this day was interpolated by assigning the rainfall that fell on Jan 1 and Jan 2 over the two day period evenly to the intervening hours.
  + For Oahu and Kauai, it was assumed no rainfall fell during that 24 hour period (due to the missing i\_rain variable). This could be adjusted on the daily aggregates if so desired.
* Data were processed on Hawaii time rather than GMT time.
  + Due to the GMT offset of 10 hours, the last year was missing 10 time steps.
  + These timesteps were filled in using the 364th day of the 2009 run to capture recent synoptic conditions.
* Day 365 for the year 2007 for Maui RCP 4.5 had negative rainfall and corrupted I\_RAIN values. This day’s values were replaced with those from day 364.
* The I\_RAIN variable was missing for Oahu and Kauai. We used the difference in precipitation between each hours. This will underestimate rainfall if more than 100 mm falls in a 1 hour period. (based on Maui data, that heavy of rainfall was a relatively rare event).
* For UDROFF for Kauai, negative values were observed for day 66 in 1994, day 1 in 1996, and days 175 and 176 for 1998. For 1994 and 1998, it looks like the counter rolled over back to 0, so the change was just marked as 0 instead of negative, and differences from the new baseline were used going forward. I did not carefully investigate the day 1 1996 negative value, but assume it is a small rounding error from the interpolation. The difference there was also set to be 0 instead of a negative value. For Oahu, the warning was just present for Jan 1 1996.
* Climatologies for cumulative variables used mean values multiplied by 365 days, so that they could be directly compared across years. This does mean that leap years are short one day’s accumulation for cumulative variables.
* Quality Control Flag: UDROFF values should be checked for the correct orders of magnitude. In particular, one year (1998) was an order of magnitude higher on average than the other years for Kauai present. Is this plausible? Does this variable have some extreme variation?
* QC outputs are missing for T2 variable (except for Kauai present). This can be an exercise for the next user to test that they can successfully use the processing scripts.

## Troubleshooting

**I got this error or something similar while downloading the data:**

**CURL Error: Failure when receiving data from the peer**

**Error in Rsx\_nc4\_get\_vara\_double: NetCDF: DAP failure**

**Var: RAINNC\_rcp85 Ndims: 3 Start: 17579,0,0 Count: 8785,64,82**

**Error in ncvar\_get\_inner(ncid2use, varid2use, nc$var[[li]]$missval, addOffset, :**

**C function R\_nc4\_get\_vara\_double returned error**

Sometimes it has memory issues while downloading. Find the spot where the download script left off and try to resume there (by running the DataDownloader manually with appropriate start/end numbers. Alternatively, you can try downloading everything again in a new session (perhaps reboot the computer to try to free up any memory that may have been tied up, or try running it with fewer other processes at the same time).

**I am running the downloader script, but it is not doing anything.**

Stop the R process, restart R, check your internet connection, and try again. Likely the internet connection was disrupted, the process hung up. If you are careful, you can adjust the start and end points to run for just the remaining files, instead of re-running for everything.

**There are small negative values in the array when I switch from hourly to daily**

This appears to be a rounding error related to the interpolation. This will be corrected at a later step, or one can take the data out of array, replace with 0’s, and then put the data back into an array format to correct it. (this was not done to avoid getting the order wrong, and there may be an easier tool for this!)

## Acknowledgements

Special thanks are due to Tom Giambelluca, Oliver Timm, Abby Frazier, Han Tseng, Kristen Sanfilippo, Katrina Fandrich, Xiao Luo, Lucas Fortini, Chunxi Zhang, and Lauren Kaiser. This research was funded by PICASC award number XXXXX.