V. Data Processing Specifications

This section outlines the process that is undertaken for each data set archived by PacIOOS. It does not include data that are not saved but merely served (*e.g.*, data that appear on the data viewer as an overlay). Data are typically downloaded and/or converted on a regular basis via an automatic cron job on *lawelawe*. Table 19 gives a listing of all these, both those that involve data transfer/convert and system crons (*e.g.*, the weekly backup). Following this, details for each data set are given.

Script	Timing	Purpose	Machine
ais_hourly	hourly, top of the hour	retrieve ship locations from AIS receiver	pae-paha
get_adcp.py	hourly, top of the hour	Get ADCP data from ACO ftp site	pae-paha
conv_HFR_kak2cdf.s	hourly, 05 past the hour	convert the raw (Matlab) KAK files to NetCDF	hfr
conv_HFR_kok2cdf.s	hourly, 07 past the hour	convert the raw (Matlab) KOK files to NetCDF	hfr
conv_HFR_kal2cdf.s	hourly, 09 past the hour	convert the raw (Matlab) KAL files to NetCDF	hfr
conv_HFR_kna2cdf.s	hourly, 11 past the hour	convert the raw (Matlab) KNA files to NetCDF	hfr
conv_HFR_kkh2cdf.s	hourly, 13 past the hour	convert the raw (Matlab) KKH files to NetCDF	hfr
conv_HFR_ppk2cdf.s	hourly, 15 past the hour	convert the raw (Matlab) PPK files to NetCDF	hfr
conv_HFR_kap2cdf.s	hourly, 17 past the hour	convert the raw (Matlab) KAL files to NetCDF	hfr
conv_HFR.s	hourly, 20 past the hour	convert the raw (Matlab) HFR files to ASCII	hfr
update_http_stats.s	daily, 03:09 AM	update AWStats	Y
get_gfs.s	daily, 10:00 AM	get and convert global forecast model (atm)	pacmod
get_gfs_pacific.s	daily, 10:30 AM	get and convert Pacific forecast model (atm)	pacmod
get_scud.s	daily, 10:00 AM	get SCUD model output	pacmod
copy_ore_output	daily, 01:30 AM	get and convert wave model output	pacmod
copy_ocn_output	daily, 01:15 PM	get ocean circ model output	pacmod
rotate_dmac.sh	daily, 2:00 PM	move model output from “today” to 7-day archive	lawelawe
conv_nss_data	daily, 04:00 PM	get and convert NSS data from DT	lawelawe
get_adp_data.s	daily, 06:00 PM	get ACO ADCP data from ftp and convert to netCDF	pae-paha
get_dhw.py	daily, 08:30 AM	get and convert Degree Heating Weeks data	pacmod

Table 19. Cron scripts run on PacIOOS server.

A. WQB-04 (Hilo) and WQB-05 (Kawaihae/Pelekane)

description: The data are supplied via ftp from YSI. They push data to the SOEST ftp server every hour, with each hourly file having a header and four lines of data (15 minute intervals). One script converts this to a regular CSV file that is then pulled into DataTurbine. Another script pulls the data from DataTurbine and makes a NetCDF file. The following is for WQB-04, but WQB-05 should be the same.

script: conv_wqb04_DT.s
timing: runs every hour at 10 past the hour
function: pulls data from SOEST ftp server and creates CSV for DataTurbine
input: ftp://ftp.soest.hawaii.edu/hioos/incoming /1518962400.csv
output: /export/lawelawe1/wqb/wqb04/for_DT/wqb04.2017-03-27_05:15:00.dat

script: conv_wqb04.s
timing: runs daily at 3:30PM
function: pulls data from DataTurbine creates NetCDF
input: https://bbl.ancl.hawaii.edu/kilonalu-data/bigisland/WQB_04/DecimalASCIISampleData/2018/02/02/WQB04_20180202000000.10.1.dat
output:
/export/lawelawe1/wqb/wqb04/netcdf_data_2018/wqb04_2018_02_02.nc

notes:

- The raw data for time=hour1 have data for hour1:15, hour1:30, hour1:45 and hour1+1:00, and these are local times
- The raw data for time=hour1 typically get posted a hour later (hour1+2:00)
- The DataTurbine files are a complete day starting at 14:00 running to 13:45 with a total of 96 lines (24*4), so they are a complete UTC day with times given as local time
- The netCDF converted files have UTC time for a complete UTC day
- The buoys used to have a pH sensor, but it was removed; we just report this as missing to keep all the files the same length
- Oxygen saturation is listed as percent and fraction. The raw data have fraction first, and percent second. The Data Turbine files switch this order, so the netCDF conversion program reads percent first (oxy1) and fraction second (oxy2)
- More generally, the columns in Data Turbine are not necessarily the same as the raw data

B. WQB-AW, WQB-KN

description: The data are supplied via ftp from YSI. They push data to the SOEST ftp server every hour, with each hourly file having a header and four lines of data (15 minute intervals). One script converts this to a regular CSV file that is then pulled into DataTurbine. Another script pulls the data from DataTurbine and makes a NetCDF file.

script: conv_wqb04_DT.s
timing: runs every hour at 10 past the hour
function: pulls data from SOEST ftp server and creates CSV for DataTurbine
input: ftp://ftp.soest.hawaii.edu/hioos/incoming /1518962400.csv
output: /export/lawelawe1/wqb/wqb04/for_DT/wqb04.2017-03-27_05:15:00.dat

script: conv_wqb04.s
timing: runs daily at 3:30PM
function: pulls data from DataTurbine creates NetCDF
input: https://bbl.ancl.hawaii.edu/kilonalu-data/bigisland/WQB_04/DecimalASCIISampleData/2018/02/02/WQB04_20180202000000.10.1.dat
output:
/export/lawelawe1/wqb/wqb04/netcdf_data_2018/wqb04_2018_02_02.nc

notes:

- The raw data for time=hour1 have data for hour1:15, hour1:30, hour1:45 and hour1+1:00
- The raw data for time=hour1 typically get posted a hour later (hour1+2:00)
- The DataTurbine files are a complete day starting at 14:00 running to 13:45 with a total of 96 lines (24*4)

C. HFR

description: The data come from each site in a Matlab binary format by Pierre's processing. The files are placed on lawelawe every hour. One script converts these binary files to ASCII for the HFR national DAC (they have their own scp access to lawelawe). Another script creates NetCDF files. Note that the first script is generic (works on all HFR sites) while the second requires a different executable for each site.

script: conv_HFR.s
timing: runs every hour at 20 past the hour
executable: lawelawe1/hfr/src/convert_HFR_mat2ascii
program: lawelawe1/hfr/src/convert_HFR_mat2ascii.F
function: reads data from radlab directories and creates CSV for HFR DAC (Scripps)
input:
lawelawe0/radlab/hioos/*site*/realtime/*yyyydddHH*00_*site*.RAD_Beam.mat
output: lawelawe/hfrnet/*site*/RDL_*site_yyyy_mm_dd_HH*0000.ruv

script: conv_HFR_*site*2cdf.s
timing: run every hour at 5, 7, 9, 11, 13, 15 and 17 past the hour (7 sites)
executable: lawelawe1/hfr/src/convert_*site*_netcdf
program: lawelawe1/hfr/src/convert_HFR_mat2ascii.F
function: accesses data from RadLab disk (.mat) and creates NetCDF
input:
lawelawe0/radlab/hioos/*site*/realtime/*yyyydddhhh*00_*site*.RAD_Beam.mat
output: lawelawe1/hfr/*site*_netcdf/*yyyy*/*mm*/RDL_*site_yyyy_ddd_HHMM*.nc

notes:

- Raw .mat files are written close to an hour after the measurement time, on the half-hour and given a UTC time in the file name.
- The programs used to only compile on an external machine (lii), but this was fixed in 2017.
- The programs read a generic input file "radar_in.mat", so the raw files are first copied to this. The syntax is different for the two however:
    - "convert_HFR_mat2ascii" with the following input: yyyy mm dd HH MM SS site. The input files have "day of year" in the title, but this program needs month and day. For example:
        - Copy 2016110010000_kal.RAD_Beam.mat to radar_in.mat
        - Day 110 is April 20th, 04/20, so run convert_HFR_mat2ascii and then enter: 2016 04 20 01 00 00 kal
    - "convert_HFR_site2cdf" with the following input: yyyy ddd HH MM SS site. For example:
        - Copy 201611001000_kal.RAD_Beam.mat to radar_in.mat

        - Day 110 is April 20th, 04/02, so run convert_HFR_kal2cdf and then enter: 2016 110 01 00 kal
- The DAC (Scripps) will occasionally purge files from the ~hfrnet directories
- Raw .mat files are kept in the RadLab directory, netcdf's in lawelawe1, .ruv are not saved

D. NSS (realtime)

description: There are two types of NSS data streams, one that comes in real-time and one that has to be manually off-loaded from the instrument and processed locally. The data come in ASCII space delineated files. The real-time ones send data straight to DataTurbine, the delayed mode ones are ASCII files that Joe puts into DataTurbine. The procedures below get the data from DataTurbine and make NetCDF files. Note that both are pulled from the archive side of DataTurbine not the Real-time ring buffer.

script: conv_nss_data
timing: runs every day at 4:00 PM
executable: lawelawe1/nss/src/new/conv2netcdf
program: lawelawe1/nss/src/new/write_netcdf.f
function: pulls hourly data files from DataTurbine and makes aggregated daily NetCDF
input: https://bbl.ancl.hawaii.edu/kilonalu-data/*loc*/…
output: /export/lawelawe1/nss/*site*/netcdf_data_*yyyy*

timing: runs daily at 3:30PM
executable: lawelawe1/nss/src/conv2netcdf
program: lawelawe1/nss/src/write_netcdf.f
function: pulls hourly data files from DataTurbine and makes aggregated daily NetCDF
input: https://bbl.ancl.hawaii.edu/kilonalu-data/*loc*/…
output: /export/lawelawe1/nss/*site*/netcdf_data_*yyyy*

notes:

- The data are reported every four minutes, and there is a single file for each hour. The times in the file are reported as local time, *i.e.*, they start at 14:00 for the previous day and run through 13:56 for the current day. The filenames on the other hand are UTC, e.g., AW02XX_001CTDXXXXR00_20180202000001.10.1.dat (Feb 02, 2018 00:00) through AW02XX_001CTDXXXXR00_20180202230001.10.1.dat (Feb 02, 2018 23:00). The program converts time to UTC.
- The delayed mode files are in UTC natively, but the filenames are not.
- QARTOD real-time QC is done on the real-time streams while delayed mode data do not have this (thus the two different programs).
- The scripts get_DT.s and conv_raw2cdf.s are used to batch process delayed mode files (typically these come in monthly).
- The Maui NSS (NS-12 and NS-13) are processed differently, and these use executable and code in their own subdirectories.
- Back when Saipan had an NSS (NS-11) this was handled with a separate program.

E. Ocean/Atmosphere model output

description: Ocean and atmospheric models are run every day and provide several day forecasts. The output files are simply copied from the common directory (shared by the data and modeling groups) as the models provide NetCDF files already. The atmospheric output we get is pre-processed by the ocean modeling group (it contains a subset of variables and is on the ocean model grid).

script: copy_ocn_output
timing: runs at 1:15 PM daily
function: copies ocean and atmosphere model output to main server disk
input: lawelawe0/dmac/0day/ocn/*.nc
output: lawelawe1/model (subdirs for ocn/atm and forecast/assimilation

notes:

- The modeling group shares a server disk called "dmac" with subdirectories for the past seven days (0day for today, 1day for yesterday, *etc*.). The files are cycled daily at 2:00 PM using rotate_dmac.sh
- The time in the WRF (atmospheric) model are sometimes incorrect so the script /root/cron_jobs/conv_wrf_time.s is run by copy_ocn_output to correct this.
- Typically the atmospheric model output is ready by 7:00 AM and the ocean model output between 10:00 AM and 1:00 PM; to ensure we get it all, the copy script is run at 1:15.
- The raw output files do not have a date in the file name, so this is added during the copy so that these files may be aggregated by THREDDS. For example, hiig_fore.nc will get copied to hiig_forec_20180223.nc for Feb 23, 2018.
- The atmospheric model output are written to lawelawe0 with a link to lawelawe1 due to limited space.

F. Wave model output

description: The wave models are run using the WRF as forcing. There are two types, a large-scale WaveWatchIII and regional SWAN grids. The WW3 output comes as ASCII files with data in rows. Each variable is in a separate file, and there are multiple grids (*e.g.*, global, Hawaii, Samoa, *etc.*). The SWAN output comes in Matlab binary files, again in different grids (*e.g.*, Kauai, Oahu, *etc.*). Two different programs are run to convert all these files to NetCDF.

script: copy_ore_output
timing: runs daily at 03:30 AM
executable: ~jimp/waves/write_*grid*_nc
program: ~jimp/waves/write_*grid*_nc.f.
function: copies WW3 output from model disk and converts to NetCDF
input: lawelawe0/dmac/0day/ww3_*grid_var.vr*
output: lawelawe1/model/ore/ww3/*grid*/ww3*grid_yyymmdd*.nc

executable: ~jimp/waves/write_*grid*_nc
program: ~jimp/waves/write_regional_nc.F
function: copies SWAN output from model disk and converts to NetCDF
input: lawelawe0/dmac/0day/ore/*grid*.mat
output: lawelawe1/model/ore/swan/*grid*/swan_*grid_yyymmdd*.nc

notes:

- The wave models run late at night/early morning (around midnight)
- Files are actually written to lawlawe2 with a link to lawelawe1 due to limited space

G. SCUD

description: The Surface Current model is run by Jan Hafner in the IPRC, and daily output is put on the SOEST ftp server. The files are already in NetCDF so the process is merely to copy these daily to the PacIOOS server.

script: get_scud.s
timing: runs daily at 10:00 AM
function: ftp data from SOEST ftp server and to PacIOOS disk
input: ftp.soest.hawaii.edu/users/hafner/SCUD/PACIOOS/*yyyy*/*yyyy-mm-dd*.nc
output: lawelawe1/model/scud/ *yyyy*/*yyyy-mm-dd*.nc

notes:

- The model is run on Jan's machine, and occasionally there are problems. If the problem is corrected, old output can be downloaded manually (email jhafner@hawaii.edu if the problem persists).

H. NCEP GFS

description: The NCEP daily forecast model is run four times per day, initialized a 00, 06, 12 and 18 hour; we only get the 12:00 run. There are many variables in the forecast of which we download and convert just a few (heat fluxes, winds, *etc*.). The raw files are accessed via an OPeNDAP call and converted to NetCDF. We get both the 0.5-degree global output and higher resolution 0.25-degree Pacific output.

script: get_gfs.s
timing: runs daily at 09:00 AM
executable: grads
program: lawelawe1/model/atm/gfs_global/src/get_gfs_0.5deg.gs
function: pulls data from NCEP OPeNDAP server and saves local NetCDF files
input:
http://nomads.ncep.noaa.gov:9090/dods/gfs_0p50/*date*/gfs_0p50_12z
output: lawelawe1/model/atm/gfs/gfs_global/data/gfs_surf_*yyyy_mm_dd*.nc

script: get_gfs_pacific.s
timing: runs daily at 09:30 AM
executable: grads
program: lawelawe1/model/atm/gfs_pacific/src/get_gfs_0.25deg.gs
function: pulls data from NCEP OPeNDAP server and saves local NetCDF files
input:
http://nomads.ncep.noaa.gov:9090/dods/gfs_0p25/gfs*date*/gfs_0p25_12z
output:
lawelawe1/model/atm/gfs/gfs_pacific/data/gfs_surf_pac_*yyyy_mm_dd*.nc

notes:

- GrADS is used to convert these files as it provides an easy OPeNDAP interface and can write NetCDF
- The scripts occasionally fail and need to be rerun. If done same day there is no problem, otherwise the date needs to be manually entered and run.
- The scripts first run grads with an argument specifying the date and time (*e.g*., grads –lcb "run get_gfs_0.5deg.gs 20180226" to specify Feb 26, 2018). The grads scripts then just gets the surface values of nine different variables and writes them to temporary (*e.g*., file01.nc, file02.nc, *etc*.) files.
- The shell script then uses ncks to concatenate the NetCDF files, ncap to correct the date (GrADS defaults to a start date of 0), and ncatted to change some of the metadata (variable long names, units, *etc*.).

I. ACO

description: At present we are accessing two data sets from the ALOHA Cabled Observatory (ACO), namely bottom pressure (of interest to the Pacific Tsunami Warning Center) and bottom currents from the ADCP. Both are high-frequency data streams.

script: aco_pressure.s
timing: runs every hour at the top of the hour
function: "listens" to port 48331 from the ACO and gets 40Hz data
input: aco-makaha.soest.hawaii.edu:48331
output: lawelawe1/aco/pressure/*time-HHMM*.acop

script: get_adp_data.s
timing: runs daily at 6:00PM
function: ftp data from ACO server and copy files to PacIOOS
input: mananui.soest.hawaii.edu:/pub/aco/adp/adp5_*yyyymmdd*.nc
output: lawelawe1/aco/adcp/yyyy/mm/adp5_*yyyymmdd_HH_HH*.nc
notes:

- The pressure data are posted immediately to ERDDAP

J. AIS

description: PacIOOS has a receiver for ship locations down at Kaka'ako (HFR site). This script "listens" for data via port 8080 and pipes the data to /export/lawelawe1/ais

script: ais_hourly.s
timing: runs every hour at the top of the hour
function: "listens" to port 8080 from the AIS receiver
input: lawelawe.pacioos.hawaii.edu:8080
output: lawelawe1/ais/aisdata/*time*.ais