Data Processing Guide

Written by David Delene: 25 July 2018
Modifications: Update 29 July 2018, 30 August 2018, 15 September 2019, 26 September 2019, 8 February 2021, 22 July 2022, 31 August 2022

Introduction

The objective of the Data Processing Guide is to provide people new to processing scientific data an overview and starting point related to material that is available. The focus is on aircraft data processing, but many software packages are useful for scientific data processing. For people new to scientific software, I recommend learning the basic concepts and structure of the Airborne Data Processing and Analysis (ADPAA) software package (Delene 2011). Updated software processing information and application is available in the more recent paper on the North Dakota Citation Research Aircraft (Delene et al. 2019). While data repository sites and documentation within the software code provides many important details, I suggest starting to use the software before trying to understand such details.

People doing scientific data processing can be divided up into two main groups; 1) Users of software packages to conduct analysis, and 2.) Developers that extend the features of software packages. One way for users to start analyzing data is to install the graphical software package Cplot2. Cplot2 is a Graphical User Interface (GUI) written as an Interactive Data Language (IDL) application (compiled binary program) that allows users to analyze files in the standard NASA-UND ASCII format. To be able to run Cplot2, the IDL run-time environment is required. An IDL license is not required but the IDL software package needs to be installed. One way for developers and advanced users to start doing data processing is to install the Airborne Data Processing and Analysis (ADPAA) package and other software packages using the Community Packages for Airborne Science (CoPAS). Upon downloading and setup of software via CoPAS, testing programs and tutorials can be utilized to learn about data processing.

Access and Accounts

Before users can software, they need access to a compute which means having an account with sufficient access. The required access and method to obtaining such access changes with every research team. Below is information related to the Aircraft team in the Department of Atmospheric Sciences at the University of North Dakota (UND).

Linux Workstations

We use Linux workstations to process data because of the multi-user features that are a fundamental part of the Linux operating system. Several people can use the computer system simultaneously. All students withing the Atmospheric Sciences Department have a University password that provides access to the Linux servers, such as "aircraft.atmos.und.edu" (Aircraft) that are used by the aircraft team at UND. However, to access the Aircraft computer system, users need an account and the correct group level file access. Typically, necessary group access is provided when accounts are setup on the Aircraft server; however, if you get a permission error you may need to be added to a group to obtain access. If you are not familiar with the Linux Operating system, the Linux Shell, and Linux text editor (for example, vi), review the information available on the Atmospheric Science Linux Wiki page, see http://wiki.atmos.und.edu/doku.php?id=atmos:linux:home.

Atmospheric Sciences Wiki Account

Wiki are great way to quickly document software related information. Users need accounts at http://atmoswiki.aero.und.edu/atmos/home. Auto account creation has been turn off on the atmospheric sciences wiki site so users accounts need be created by an administrator (for example, David Delene) and user need to be in the "members" group.

Repositories (Sourceforge/Github)

Sourceforge and Github are two software repositories that provide free accounts. To be a developer on ADPAA, users need a Sourceforge account. After you create a Sourceforge account, send your username to a project administrator (for example, David Delene for ADPAA) to be added to the software repository as a developer, which enable you to upload software changes. Additionally, Sourceforge provides a Wiki for projects, which users need to be access to allow modification of pages, requires the Wiki Administrator to create their accounts and provide "Editor" permission. In addition to Sourceforge, people should create a Github account so they can be added as a developers on CoPAS.

Zotero Account

When writing articles and student thesis, a reference manager is very useful to save time on creating the reference section. Zotero is an cloud computing based system that enables management of references, which work on many platforms, including Linux. Hence, people should create an Zotero account and a group that is shared with their adviser and co-authors (for example, David Delene).

Working on a Linux Workstations

Typically, there are two methods of accessing a Linux workstation: shell access and the full graphical interface access. To access a Linux workstation shell, secure shell access is provided by the ssh server and an ssh client. Putty is a ssh client for the Windows Operating System, see http://wiki.atmos.und.edu/doku.php?id=atmos:citation:soft:putty_ssh, that provides access to a Linux shell. To obtain full graphical interface access, user either need to physical be at the Linux Workstation or have remote access the workstation using a Virtual Network Computing (VNC) client, see http://wiki.atmos.und.edu/doku.php?id=atmos:citation:soft:vnc. Most tasks can be done via a Linux Shell; however, while it is complex to setup, VNC provide a more robust graphical interface that is faster to access graphical programs.

Software

New users can quickly start looking at data using the Cplot2 visualization program. The Cplot2.sav file is available for download on the ADPAA Sourceforge site. There is a Cplot guide available on the ADPAA Wiki site. The Community Packages for Airborne Science (CoPAS) is a Github project designed to facilitate the installation, setup, and integration of open source software and packages related to cloud physics, in-situ airborne data. CoPAS is utility design to make it easier to get started as a developer of community software. The Airborne Data Processing and Analysis (ADPAA) package is an open-source software package containing a collection of programs and scripts to process and analyze data from in-situ instruments deployed on airborne platforms. The ADPAA package was started to process data on the North Dakota Citation Research Aircraft but has been used to process data on many airborne platforms. There is a Wiki page that documents how to begin to use ADPAA, describes the files used by ADPAA, provides guides, and gives details on many instrument processing streams. The Airborne Data Testing and Evaluation (ADTAE) project develops open source resources to test and evaluate software used to process and analyze measurements from scientific instrument deployed on airborne platforms. To start processing data, use data available from your field project or ADTAE to process data using a field project script.

Several users can access a Linux workstation and run the same software; however, due to authorization restrictions of repositories, only individual, local copies of the repository can be updated. The Aircraft Server has airborne software installed under /usr/local/CoPAS, along with environmental variables defined in /etc/profile.

- Test ADPAA install, within a shell windows, type “cplot2”
- Also, should be able to run new version using “aplot”

Developers and power users will want to setup their own software within their home directories. Getting the software setup has been made easier by development of scripts to installing and testing software. The following are the steps required to install CoPAS, process some data image array probe data, and examine the results on a Linux computer system.

- Clone CoPAS [$ git clone https://github.com/daviddelene/CoPAS.git]
- Test CoPAS scripts [$ cd CoPAS && CoPAS.py -t]

There is a python 3 (CoPAS.py) version of the CoPAS script and a python 2 version (CoPAS_python2.py). The -t option provides a way to ensure that necessary support software is available. Using Python virtual environments is a best practice to isolate project-specific dependencies and create reproducible environments. In other words, it’s a way to avoid conflicting dependencies that lead to dependency hell (see https://developers.redhat.com/blog/2018/08/13/install-python3-rhel/). The CoPAS repository contains the requiremens_CoPAS.txt file that provides the python package requirements for the Python 3 version of CoPAS. The README.md file contains information on cloning CoPAS, setting up the python environment, and using CoPAS to download packages.

ADPAA Setup and Install ADPAA For Developers of ADPAA

Note that items after $ are command to be enter into shell terminal window.

Clone CoPAS:

$ git clone https://github.com/daviddelene/CoPAS.git

Test for Required Packages:

$ cd CoPAS && ./CoPAS.py -t

If test shows missing packages, setup the Python3.6 environment, see README.

Setup to install ADPAA with Source Code using CoPAS:

The setup process of defining SOURCEFORGE_USER and SVN_USERNAME are for ADPAA developers. For users or people without Sourceforge ADPAA access, or users that don't to setup the system to commit code changes as developers, see the next section for setup and install information. However, for ADPAA developers, do the following:

Set username in ~/.bashrc using environmental variables.
Add the following lines to ~/.bashrc. Where ~ mean your home directory

```
SOURCEFORGE_USER='name'
SVN_USERNAME='name'
export SOURCEFORGE_USER SVN_USERNAME
```

Where 'name' is your developer's username on SourceForge. For example using username ltwohey7:

```
SOURCEFORGE_USER='ltwohey7'
SVN_USERNAME='ltwohey7'
SOURCEFORGE_USER SVN_USERNAME
```

After adding the above environmental variables source .bashrc

```
$ source ~/.bashrc
```

To test that environmental variables are set

```
$ echo $SOURCEFOGE_USER
```

To install ADPAA

```
$ cd ~/CoPAS
./CoPAS.py ADPAA
```

Setup ADPAA:

To setup the ADPAA software, including defining Paths, execute the following from the command line:

```
$ ~/CoPAS/ADPAA/bin/adpaa -s
$ source ~/.bashrc
```

Test that ADPAA is working:

```
$ aplot
```

User’s Tutorial (For Users of ADPAA)

Clone CoPAS:

$ git clone https://github.com/daviddelene/CoPAS.git

Test for Required Packages

$ cd CoPAS && ./CoPAS.py -t
If test shows missing packages, Setup the Python3.6 environment, see README.

Install ADPAA, ADTAE, and SODA:

$ ./CoPAS_python2.py ADPAA ADTAE SODA

Setup ADPAA:

$ ADPAA/bin/adpaa -s

Install ADPAA:

$ source ~/.bashrc

Test that ADPAA is working:

$ cplot2
Quit cplot2 by selecting the "Quit" button.

Run CIP Process:

$ cd ADTAE/TestPrograms
$ ./test_process_soda2.cip.bash

Processed Files are located in ~/TESTING/FlightData/20170904_095807/PostProcessing

$ cd ~/TESTING/FlightData/20170904_095807/PostProcessing

Create Images:

$ mkdir CIP_Images
$ cd CIP_Images
$ plotcip start = 1 png = 1 timingbars = 1 ../17_09_04_09_58_48.cip

Important Images are 17_09_04_15_11*.png

$ display 17_09_04_15_11_28.5099.cip_image.png
$ gnome-open 17_09_04_15_11_28.5099.cip_image.png

Figures (September 4, 2017)

Below are image from the NASA P3 on the return ferry flight for ORACLES2 (Figure 1). Figure 2 and 3 show the that data was obtained over most of the flight and the temperatures got down to 12 °C. There was no 2D-S or HVPS3 data due not being able to run the instruments due to fuel leak on the aircraft.

Figure 1: Image showing the NASA P3 aircraft flight track on September 4, 2017 from Barbados to Wallops Island.

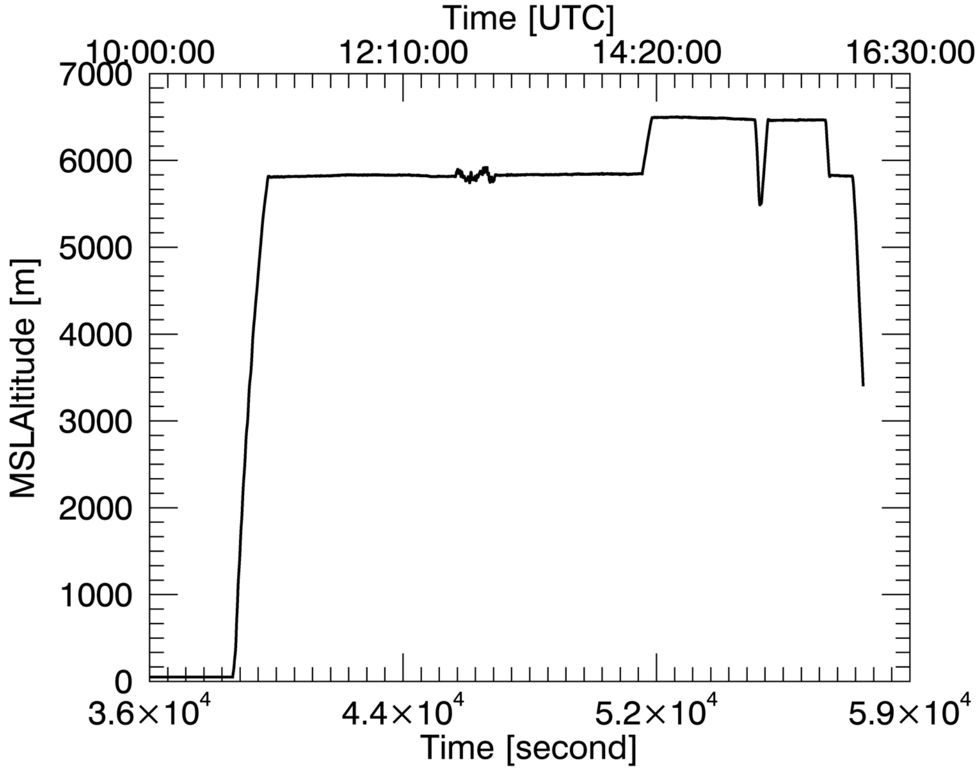

Figure 2: Image showing the NASA P3 aircraft altitude on September 4, 2017 during ferry flight from Barbados to Wallops Island.

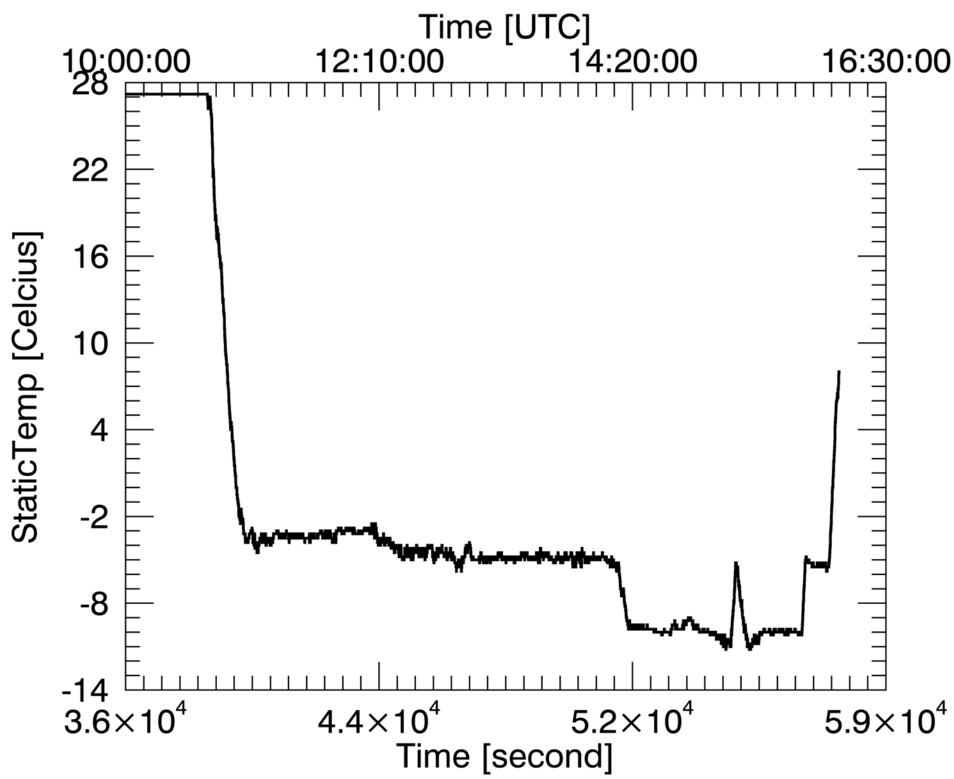

Figure 3: Image showing the NASA P3 aircraft temperature on September 4, 2017 during ferry flight from Barbados to Wallops Island.

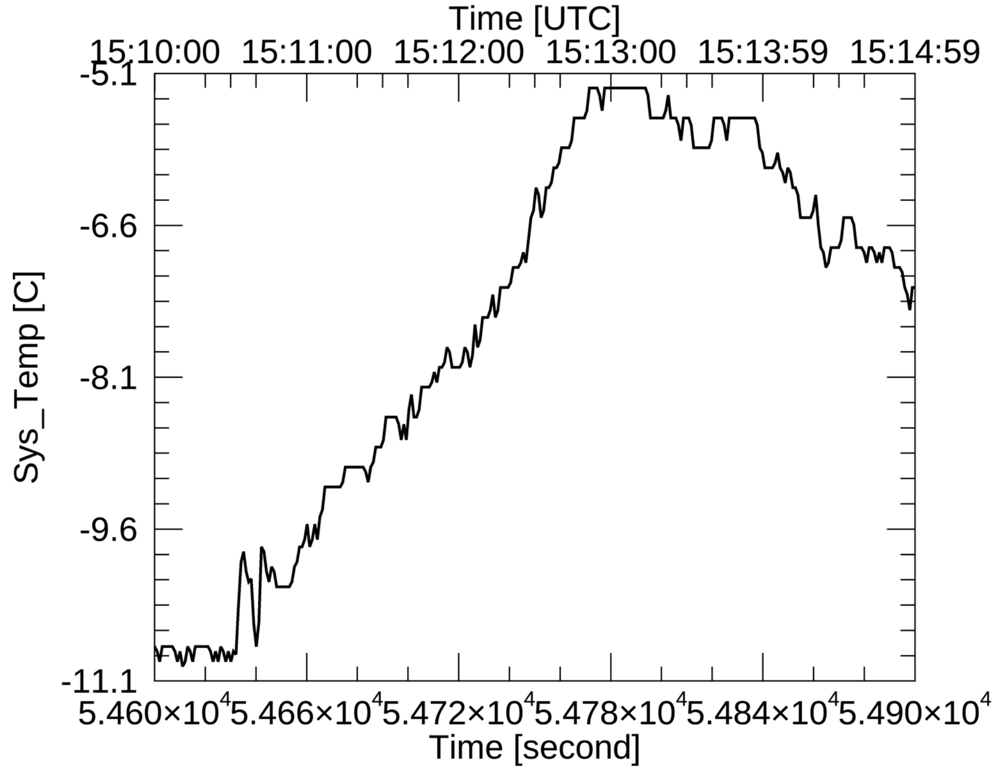

Figure 4: Image showing the NASA P3 aircraft temperature on September 4, 2017 from 15:10 to 15:15 UTC.

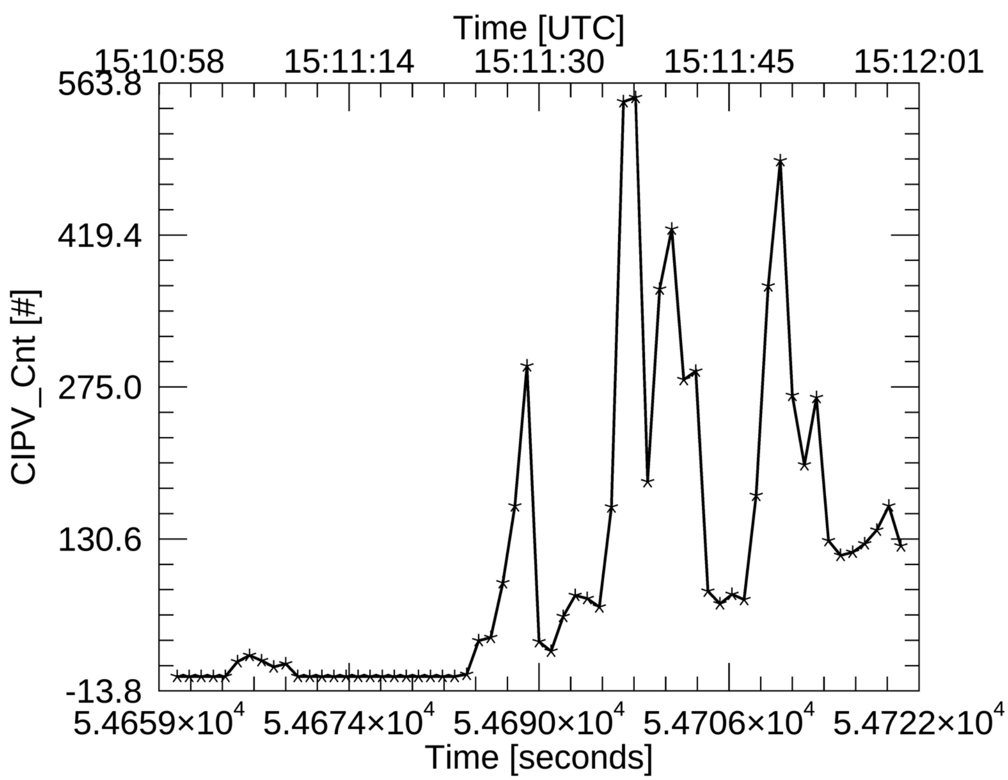

Figure 5: Image showing the NASA P3 aircraft cloud particle counts from the Cloud Particle Imager (CIP) on September 4, 2017 from 15:10 to 15:15 UTC. The CIP diode size is 25 um and the CIP has a total of 62 diodes (2 used for end elements).

Conclusions

While reviewing existing material, people are encouraged to make notes on missing information and how to better organize the documentation. One of the first contribution for a new user can make is to update the Wiki page documentation or write a new tutorial. Creating documentation and tutorial is a good way for new people to contribute to analysis of data using the Cplot2 program. For processing data, people can contribute by developing new programs and modifying existing programs, typically under the direction of a package administrator or experienced developer.

References

Delene, D. J., 2011: Airborne data processing and analysis software package. *Earth Sci. Inform.*, **4**, 29–44, https://doi.org/10.1007/s12145-010-0061-4.

——, K. Hibert, M. Poellot, and N. Brackin, 2019: The North Dakota Citation Research Aircraft Measurement Platform. *SAE Tech. Pap. 2019-01-1990*, https://doi.org/doi: 10.4271/2019-01-1990.