

SIMPLE MINIMALISTIC OFFLINE PLOTTING OF CADI DATA FILES

Per Ivar Emanuelsen¹

Chris Hall¹

Magnar G. Johnsen¹

¹Tromsø Geophysical Observatory, UiT – The Arctic University of Norway

Introduction

A critical element in any observational program is to remotely monitor instrument health and performance. Simple real-time plots provide this type of information and in addition they contain valuable space weather information for nowcasting and acquisition of situational awareness.

Several software packages exist for plotting and scaling (analysis) of ionospheric soundings [1,2] using the Canadian Advanced Digital Ionosondes (CADIs) [3]. However, a simple method, not entailing installation of a full analysis package, for generating ionograms offline, or for near real-time displays generated on central server, is not easy to locate. Using information in the CADI documentation [4] combined with that found in analysis-package source-code [specifically 2], we have constructed a minimal Python package that takes a standard CADI output file [4] and generates (for example) a corresponding ionogram as a .png file. Moreover, written in Python, no licensing is required.

In addition to plotting of individual classic ionograms, we offer plotting in the form of frequency profiles versus time and virtual height – essentially stacked ionograms with colour-coding indicating critical frequency, although not distinguishing between O and X modes. This displays the temporal development of ionospheric features, such as persistence and diurnal variation. Operating on an entire directory, the program can both display historical data and display continually updating files so as to show results in near real-time.

Method: individual ionograms

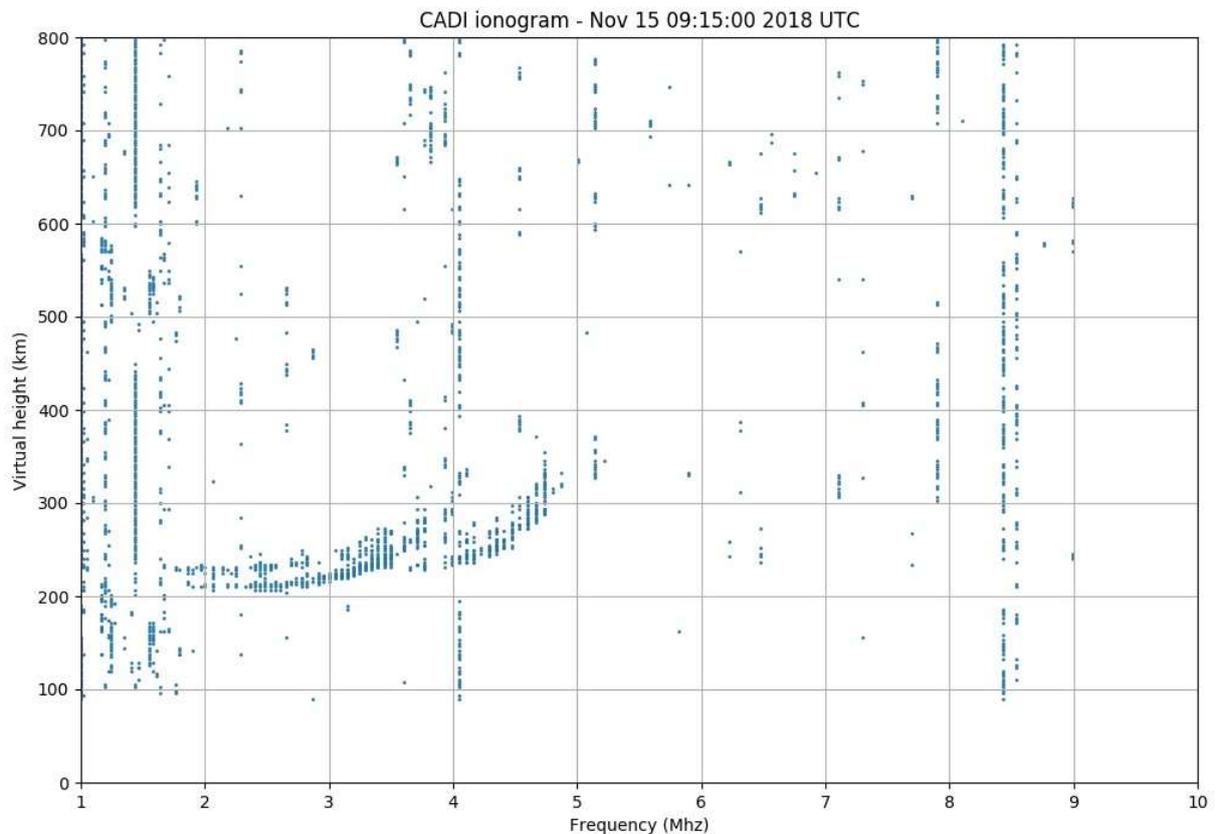
The code is written for python ≥ 3.6 , but might also work for earlier versions above 3.0.

The code will not work for python 2.7.

Additional requirements are numpy (<https://numpy.org>) and matplotlib (<https://matplotlib.org>).

usage: `python rdatafile.py inputfile outputfile`
inputfile = file name of the CADI data input file
outputfile = file name of the resulting plot

A typical output is:



outfile should have extension `.png` to produce png-files or `.pdf` to produce pdf-files.
Supported formats `eps`, `pdf`, `pgf`, `png`, `ps`, `raw`, `rgba`, `svg`, `svgz` (depends on version of matplotlib)

The bottom part of the python code can be changed to direct output to the screen or a file or both, thus:

- to toggle output to file comment/uncomment the following two lines:
 - `savefilename = sys.argv[2]`
 - `fig.savefig(savefilename)`
- to toggle output to screen, comment/uncomment the following line:
 - `plt.show()`
- the usage when output to file is disabled and output is to the screen only:
 - `$ python rdatafile.py inputfile`

The `rdatafile.py` distribution and any ancillary information (including data for testing purposes) is at GitHub:

<https://github.com/pi-tgo/rdatafile>

Method: temporal development (“stacked” ionograms)

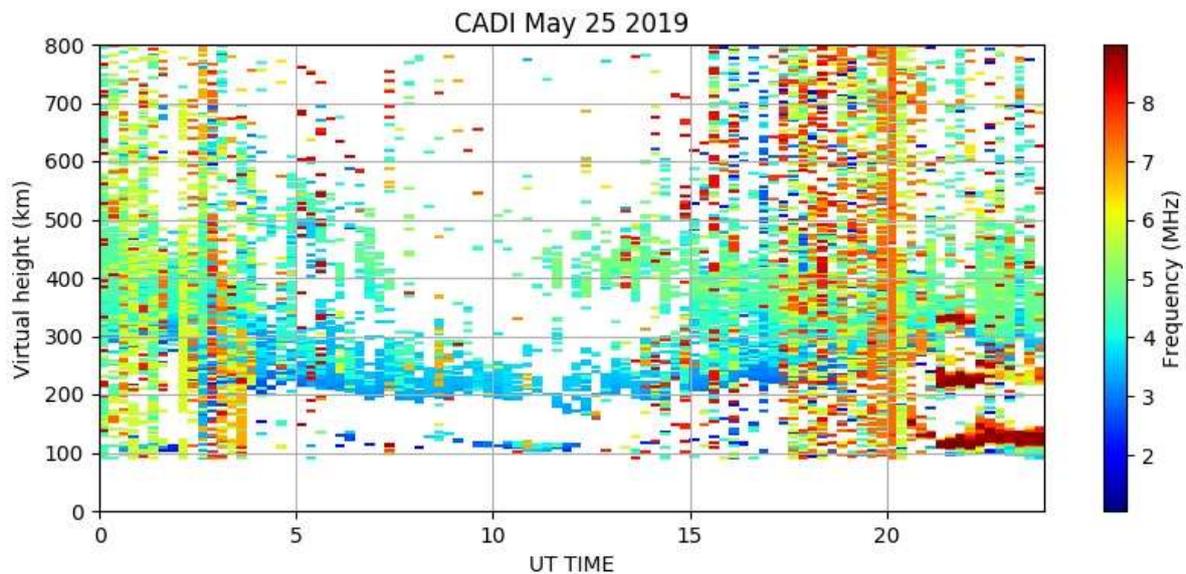
The code is written for python ≥ 3.6 , but might also work for earlier versions above 3.0.

The code will not work for python 2.7.

Additional requirements are numpy (<https://numpy.org>) and matplotlib (<https://matplotlib.org>).

usage: python *cadi24h.py sourcedir outputfile*
sourcedir = directory containing a single day of type .md49 CADI data files
outputfile = file name of the resulting plot

A typical output is:



outputfile should have extension .png to produce png-files or .pdf to produce pdf-files.
Supported formats eps, pdf, pgf, png, ps, raw, rgba, svg, svgz (depends on version of matplotlib)

The bottom part of the python code can be changed to direct output to the screen or a file or both, thus:

- to toggle output to file comment/uncomment the following two lines:
 - `savefilename = sys.argv[2]`
 - `fig.savefig(savefilename)`
- to toggle output to screen, comment/uncomment the following line:
 - `plt.show()`
- the usage when output to file is disabled and output is to the screen only:
 - `$ python cadi24h.py sourcedir`

The *cadi24h.py* distribution and any ancillary information (including data for testing purposes) is at GitHub:

<https://github.com/pi-tgo/cadi24h>

References and literature URLs

- [1] Autoscala software Pezzopane, M., Scotto, C., Tomasik, Ł. et al. Acta Geophys. (2010) 58: 513. <https://doi.org/10.2478/s11600-009-0038-1>

- [2] IDL software Grant, I. (1995), <http://chain.physics.unb.ca/chain/pages/software>, Univ. Western Ontario.
- [3] CADI product <https://www.sil.sk.ca/content/cadi>
- [4] CADI users manual https://www.naic.edu/~phil/hardware/cadi/Cadi_users_Manual.pdf

Contact person(s)

Per Ivar Emanuelsen, email: per.i.emanuelsen@uit.no

Acknowledgements

Thanks to Chris Meek (Univ. Saskatoon) for directing us to the IDL software distribution, Cato Scotto (INGV), David Themens (Univ. New Brunswick) and many others (our apologies for not naming everybody)