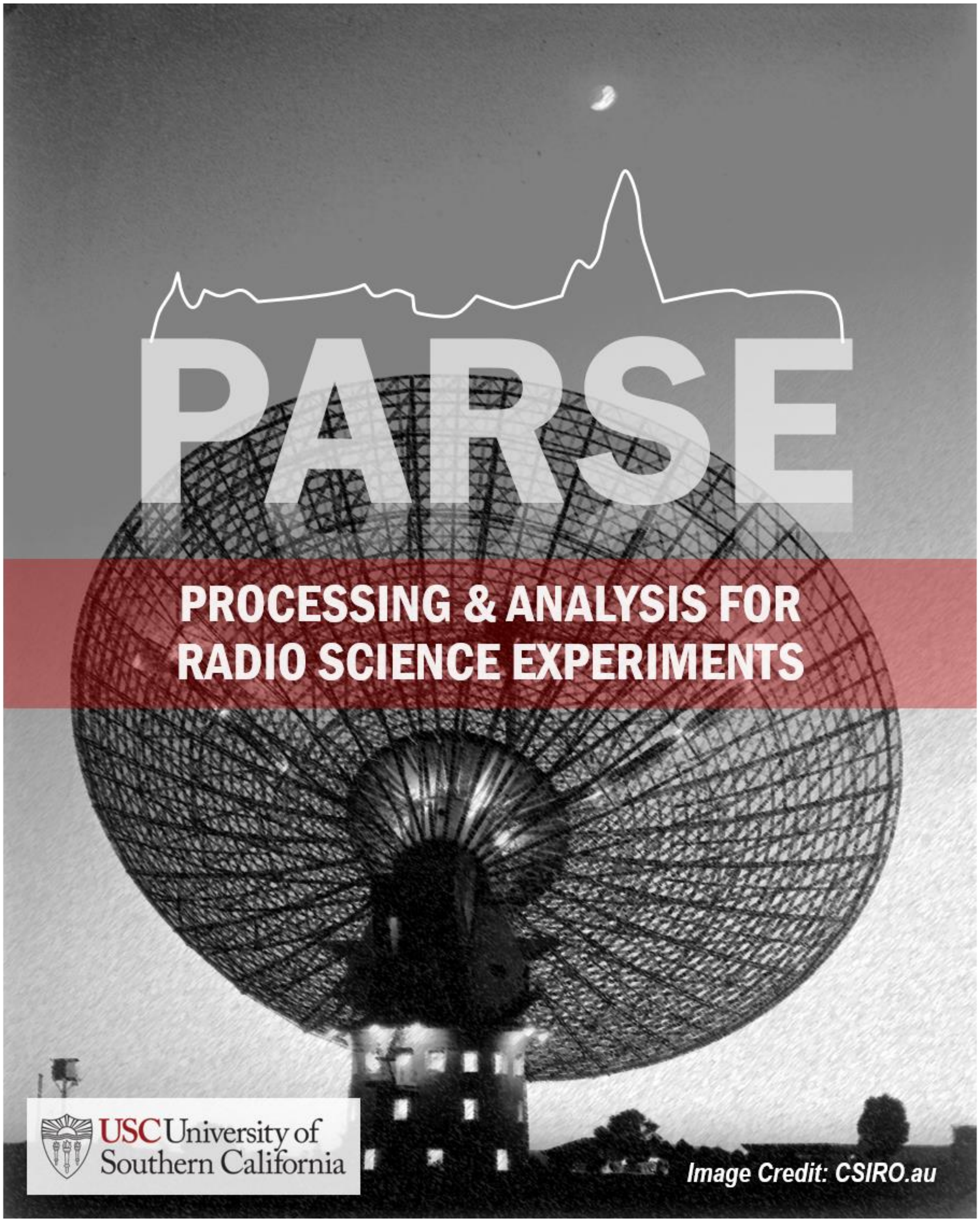
(rev. 08 Nov 2021)



User's Guide (v1.0)

Elizabeth M. Palmer | Paul Sirri | Essam Heggy

Contents

Introduction to PARSE

PARSE is a user-friendly GUI tool to assist planetary scientists in analyzing Deep Space Network (DSN) raw radio science datasets, without requiring expertise in signal processing techniques. PARSE can be used on bistatic radar (BSR) surface-scatter experiments, which use the radio communications antenna aboard a spacecraft to transmit X- or S-band radiowaves that scatter from the planetary object's surface and are then received by the DSN (Figure 1). An example of such an experiment is detailed by Palmer, Heggy & Kofman (2017) for the Dawn mission at Asteroid Vesta.

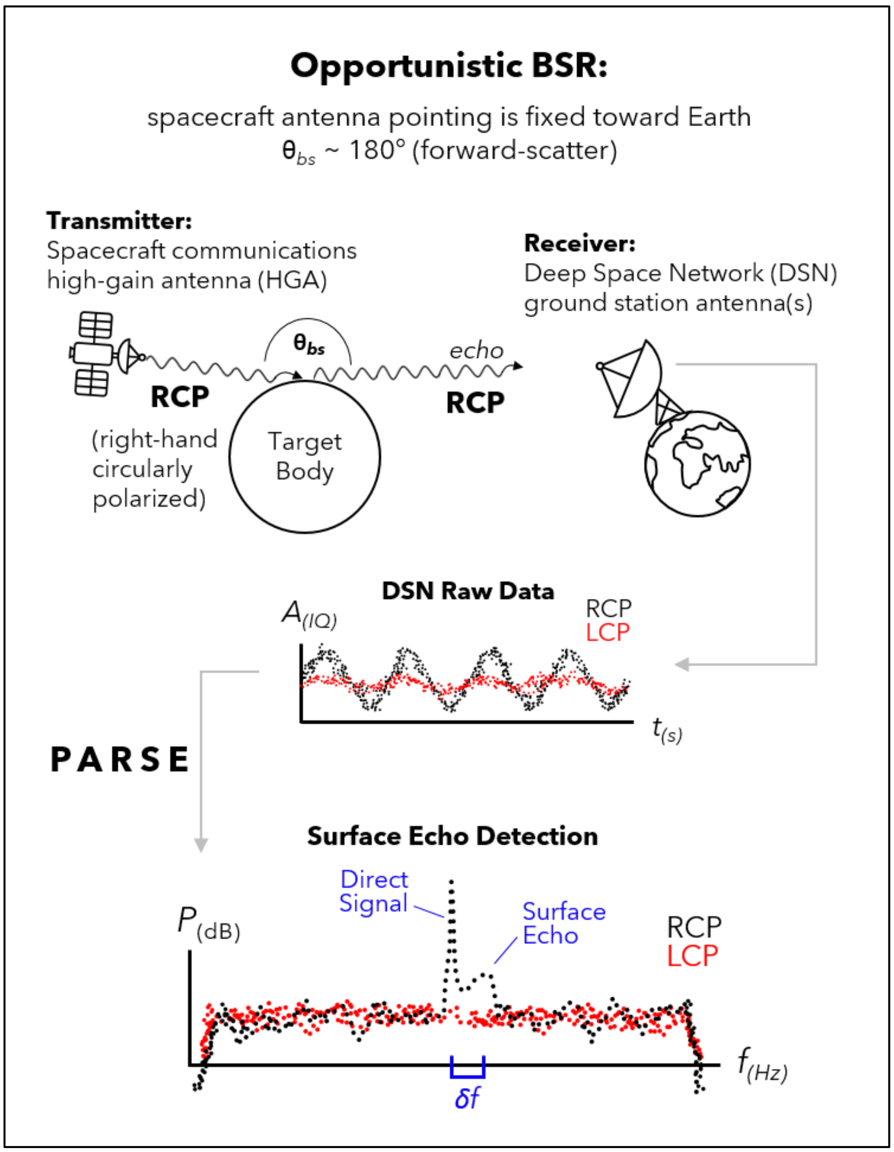


Figure 1. Acquisition geometry of an opportunistic bistatic radar (BSR) experiment. PARSE takes in the raw amplitude time-series data and automates the processing chain that generates power-frequency plots, which are used to measure the strength and bandwidth of surface-scattered echo signals.

The nominal use-case for this first release of PARSE is an opportunistic BSR experiment such as that in Figure 1, whereby the spacecraft's antenna-pointing remains fixed toward Earth, such that any surface-scatter echoes occur at grazing incidence when the spacecraft occasionally passes behind—or emerges from behind—the target body (i.e., an occultation). In this forward-scatter geometry, where the bistatic angle θ_{bs} approaches 180°, the handedness (whether right- or left-) of the circular polarization of the transmitted signal is preserved in the surface-scattered echo (shown in Figure 1 for the case of right-circular "RCP" transmission).

PARSE calculates the expected frequency separation (i.e., differential Doppler shift δf_{calc} as shown in Figure 2) between the direct signal and surface echoes for this grazing-incidence geometry (employing the algorithm from Palmer & Heggy, 2020), and uses this value to calculate signal processing parameters (such as frequency resolution of the output plots) to ensure that the user can distinguish the surface echo from the direct signal.

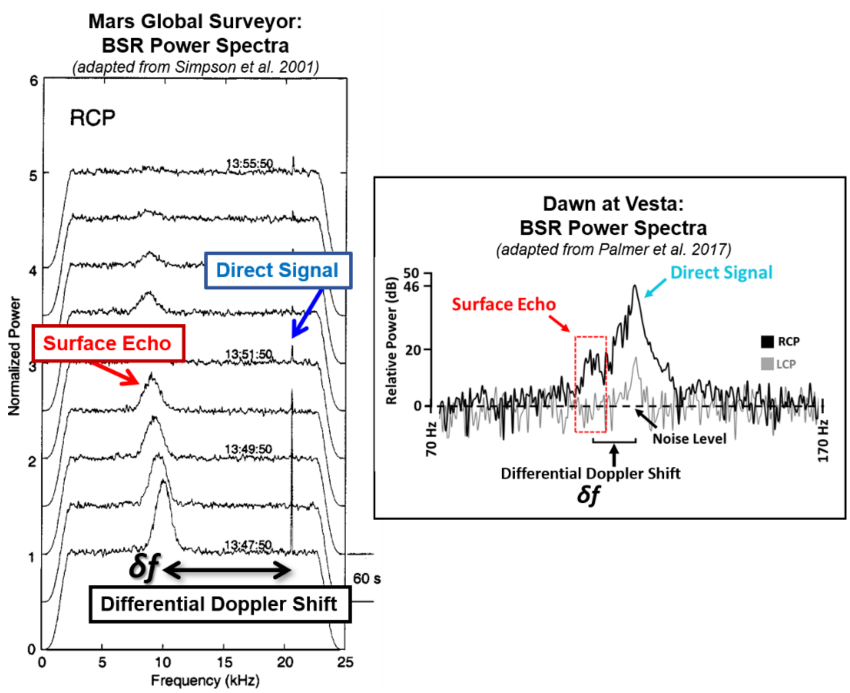


Figure 2. The differential Doppler shift δf is an important parameter calculated by PARSE for a given mission and target body. This calculated frequency separation δf_{calc} between the direct signal and surface echoes will determine the optimal parameters for signal processing so that the user will be able to identify and quantify the strength and shape of surface echoes and distinguish them from the direct signal.

Future versions of PARSE will be expanded to include full support for detecting surface echoes from BSR surface-scatter observations in the backscatter regime, which have lower angles of incidence (0° < θ_{bs} < 180°). In this observation geometry, the surface-scattered echo can be distinguished from the direct signal by having both RCP and LCP components, unlike the forward-scatter case (e.g., Simpson *et al*. 2011).

PARSE TOOL REFERENCE MATERIALS:

Peer-reviewed article:

- Sirri, Palmer & Heggy [Under Review: *The Planetary Science Journal*]. "Processing and Analysis for Radio Science Experiments (PARSE): Graphical Interface for Bistatic Radar."

Video tutorial (quick run-through):

- https://youtu.be/JcRaaFpzjIg

Documentation:

- https://github.com/PARSE-team/PARSE/blob/main/README.md
- This guide: https://github.com/PARSE-team/PARSE/blob/main/UsersGuide_v1__08-Nov-2021.pdf

Source code:

- https://github.com/PARSE-team/PARSE/tree/main/src/main/python

Getting Started

System Requirements

- Microsoft Windows or MacOS X
- 6 GB free space on hard-drive (large size is due to bundling of PARSE application with large example datasets)

Installation Steps

1. Downloading PARSE:

Navigate to our GitHub repository: https://github.com/PARSE-team/PARSE

1. In the column on the right-hand side of the repository homepage, click "Releases". (See Figure 3)

2. Select a release, then click on "Source code (zip)" to download the project repository.

   Note: The download is large (6-7 GB) since it includes raw data files, so it will take time to download depending on your connection speeds.

3. After unzipping the file, look inside the "PARSE" directory to find the installer file for your system and run it. Due to the size of the bundled data files, please expect 1-2 minute delays when opening the application or its installer for the first time.
   a. For Mac OS X: navigate to "PARSE-1.0 / build / mac / PARSE.dmg"
   b. For Microsoft Windows: navigate to "PARSE-1.0 / build / windows / PARSESetup.exe"

      Note: If using Microsoft Windows, the installer may display a warning message. If so, click "More Info" and "Run Anyway" to begin the installation.

4. Once the installation is completed, PARSE will be available in your applications folder. Due to the size of the bundled data files, please expect 1-2 minute delays when opening the application or its installer for the first time.

5. After opening the application, use the following menu options to learn about how PARSE works. Some of these resources have been linked here for your convenience.
    a. Tutorial Video (https://youtu.be/JcRaaFpzjlg) : a brief illustrative demonstration of PARSE for new users
    b. User's Guide (https://github.com/PARSE-team/PARSE/blob/main/UsersGuide_v1.pdf) : the official documentation for PARSE
    c. Relevant Publications: publications that discuss PARSE's underlying pipeline

Developers: To begin working with source code directly, install Git LFS on your device before cloning this repository.

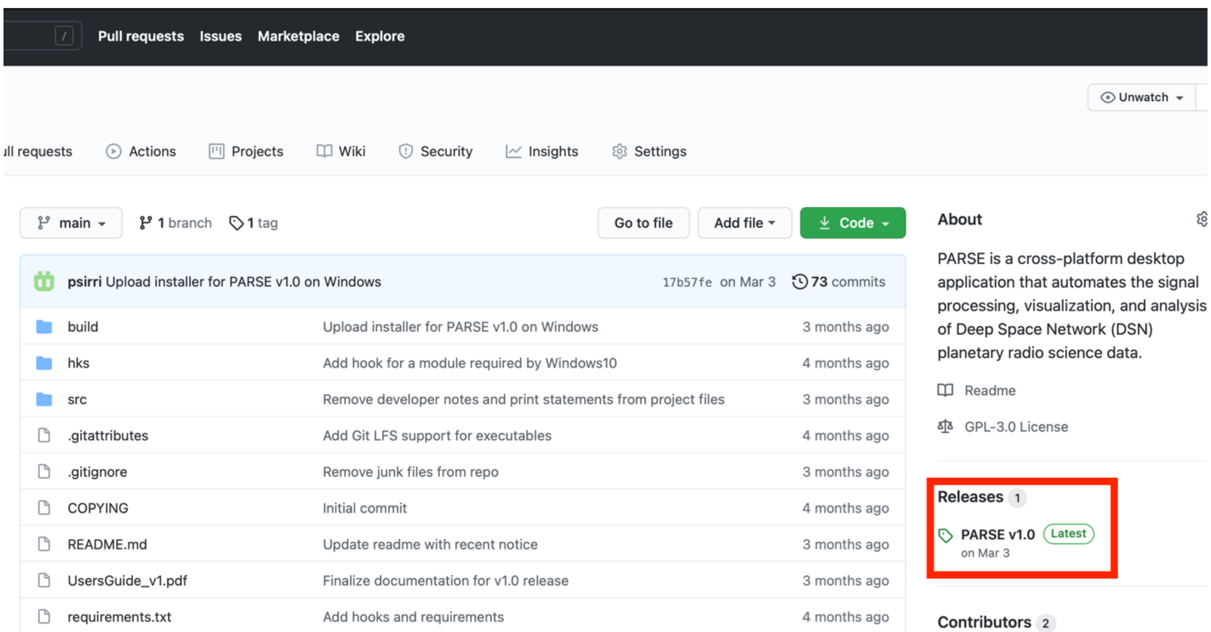


Figure 3. Screenshot of where to download the latest version of PARSE from the GitHub repository.

2. Installation:

Once downloaded, click the .exe or .dmg file to begin the installation process. This may take as long as 5-10 minutes (depending on your CPU speeds) since there are pre-bundled example datasets included along with the GUI itself.

Note: Because this is an unsigned application, Windows systems may generate pop-ups warning about the safety of the file. Microsoft will warn you that it prevented the application from installing, but click “More Info” and “Run Anyway” to acknowledge that you trust the developer and continue with the application’s installation process.

3. **Running PARSE:**

Navigate to the build folder within the directory that you just installed on your system and click on the application called “PARSE”.

Note: PARSE will start up slowly the first time it is run, again due to the large datasets pre-bundled with this release (~45 seconds on an i9 processor).

4. **Other Tips & Suggestions:**

- After opening PARSE for the first time, we recommend watching the video tutorial to see a quick run-through of how the GUI is used (click the “Video Tutorial” button on the Start Screen as shown in Figure 5)
- There will be a delay of a few seconds after clicking the “Apply Changes” button when you reach the GUI’s Signal Processing window (~3 seconds on an i9 processor).
- When choosing a user-defined dataset for the first time, PARSE will run slowly due to the time-consuming process of converting a large ASCII file into a compact Python-readable binary format. See also page 16 of this document.

Workflow and Screenshots

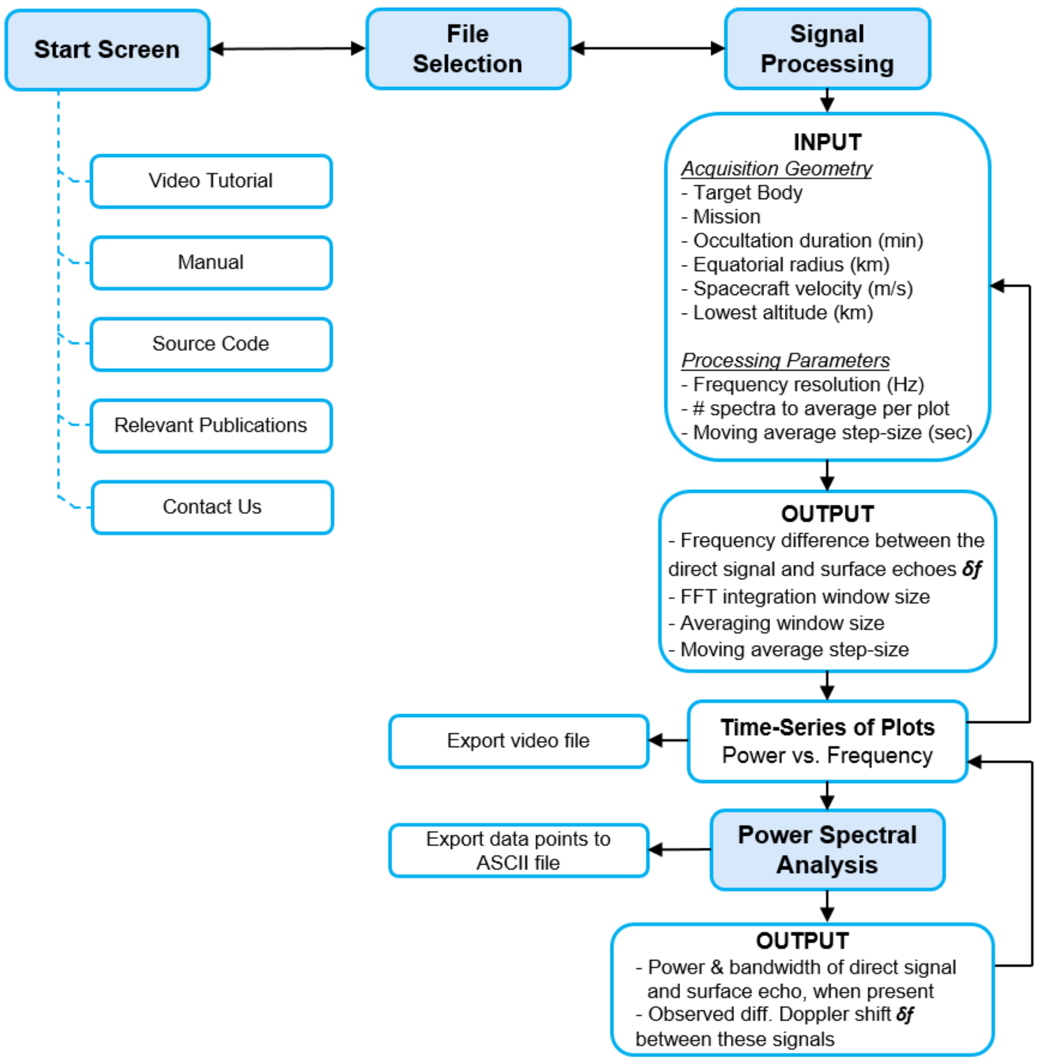


Figure 4. Block diagram showing an overview of how PARSE works.

Once the application is launched, the user will see a start screen with several buttons in the right-hand column (Figure 5). After clicking one of the three center buttons to select the type of input data (Dawn, Rosetta or User-Defined), a file selection dialog window will pop up and the user will choose the directory containing the desired data.

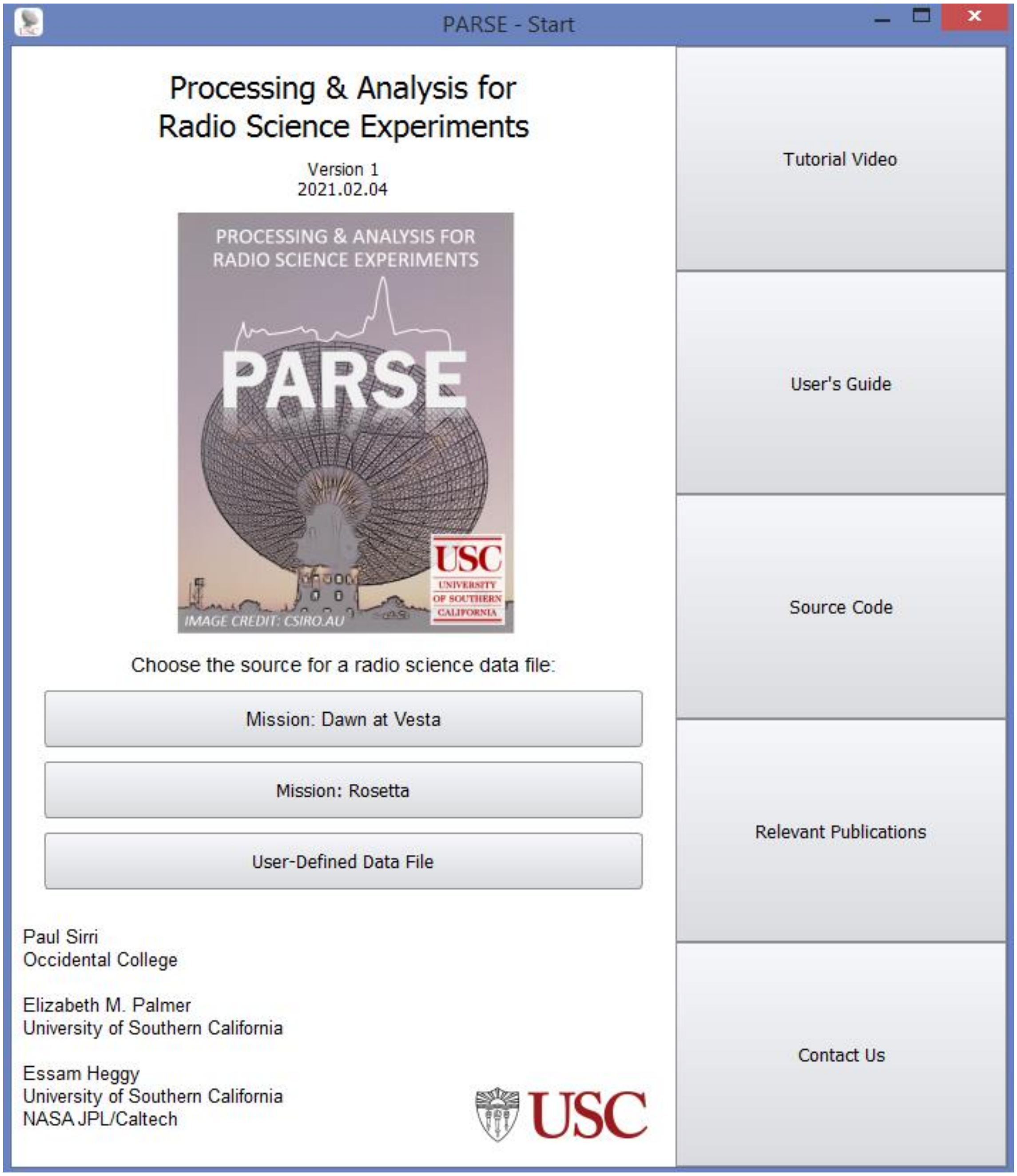


Figure 5. Start screen of PARSE GUI.

Next, the user is brought to the GUI's file selection window (Figure 6) showing a list of available files on the left, and any header information on the right (taken from the detached PDS3 label file when available). The user should highlight a pair of files—one with "RCP" data (i.e., right circularly polarized) and one with the matching "LCP" data (left circularly polarized) that was acquired at the same time. Then click "Process."

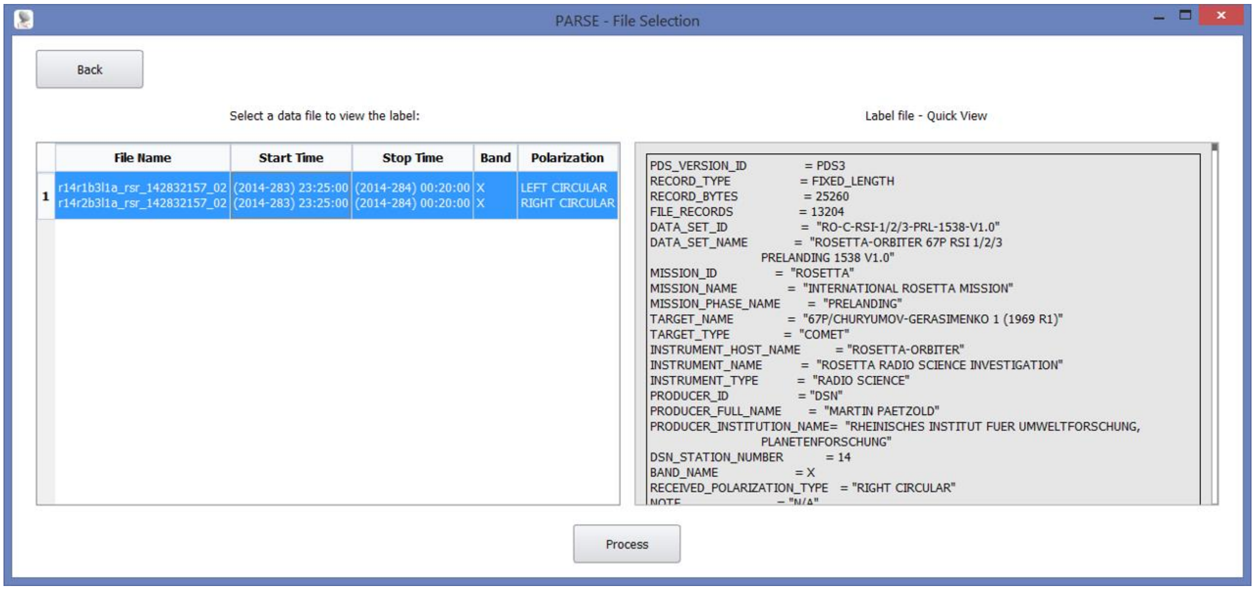


Figure 6. Second screen (file selection) of PARSE GUI. If there is an accompanying PDS3 label/header file with the data file, it will display as a preview on the right.

Next is the Signal Processing window (Figure 7). The left panel displays all the input parameters–some adjustable, others fixed–grouped by "Acquisition Geometry" (spacecraft and target information), "Radio Data Analysis" (signal processing parameters estimated from the latter), and "Window Properties" for the output plots.

On the right are two plots: the large upper one is a playable video that goes frame by frame through each of the processed power spectra; the lower plot is an overview of the signal's peak strength over time. As the video is played, a red bar scrolls through the overview plot.

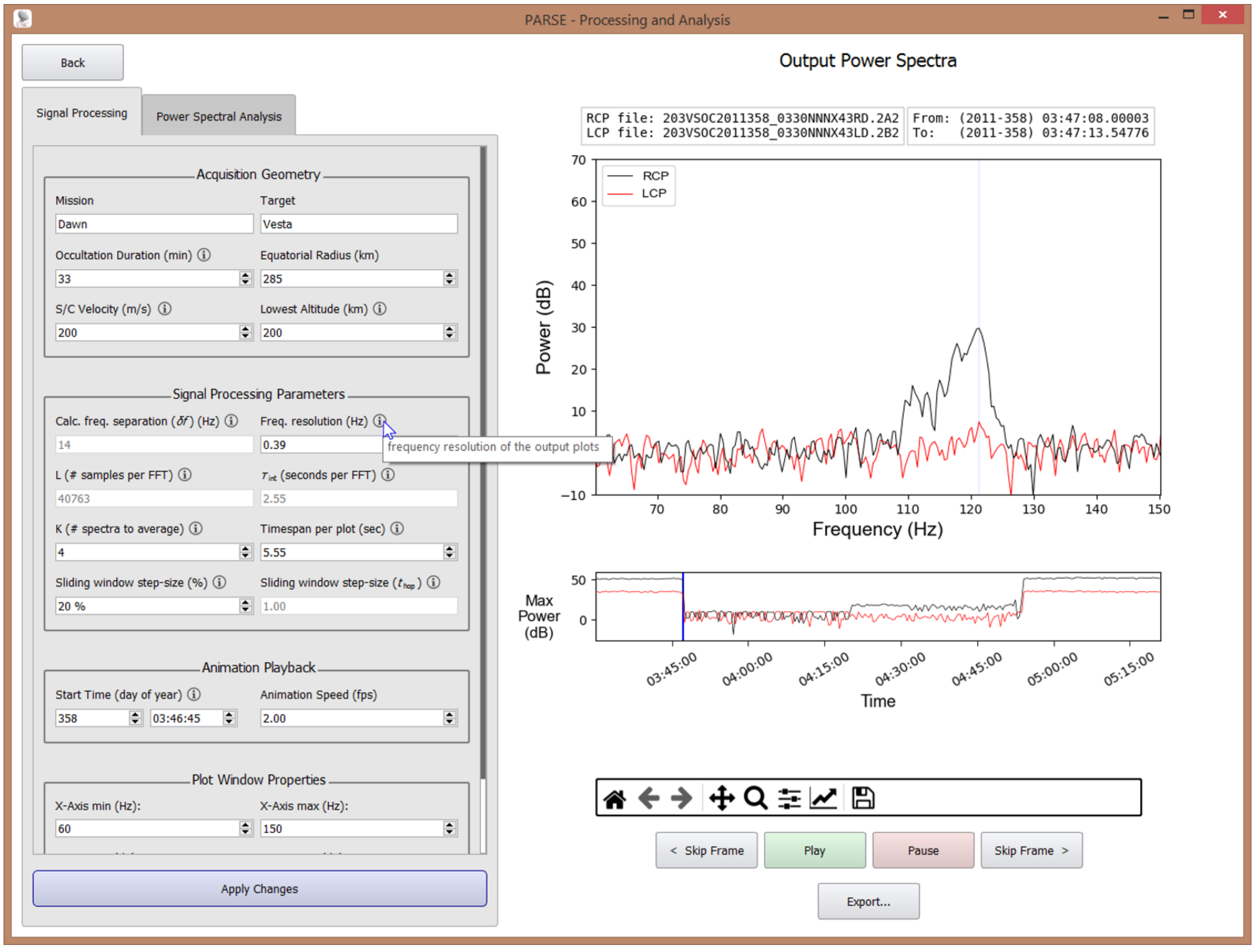


Figure 7. Third screen (signal processing) of PARSE GUI. Computes suggested processing parameters and then automates the processing chain for plotting the signal power vs. frequency over time.

When the user identifies a particular frame (plot) of interest, e.g., one containing a potential 'echo' signal, the user should click the "Spectral Analysis" tab just above the input parameter list to be brought to the fourth and final window of the GUI (Figure 8).

Here, the user will be able to measure the bandwidth, peak power and observed frequency separation (δf_{obsv}) of the direct signal and echo signal (when present) by selecting the x-range of frequencies where the echo is thought to be contained (see the "Selected Range" bounded in blue in the large plot on the right in Figure 8).

The x-range is selected by changing the "x-axis min (Hz)" and "x-axis max (Hz)" values in the left panel under the "Settings" group. Clicking the "Refresh Plot" button will update the plot shown on the right, as well as update the output values displayed in the left-hand panel under the "Results" > "Selected Range" section.

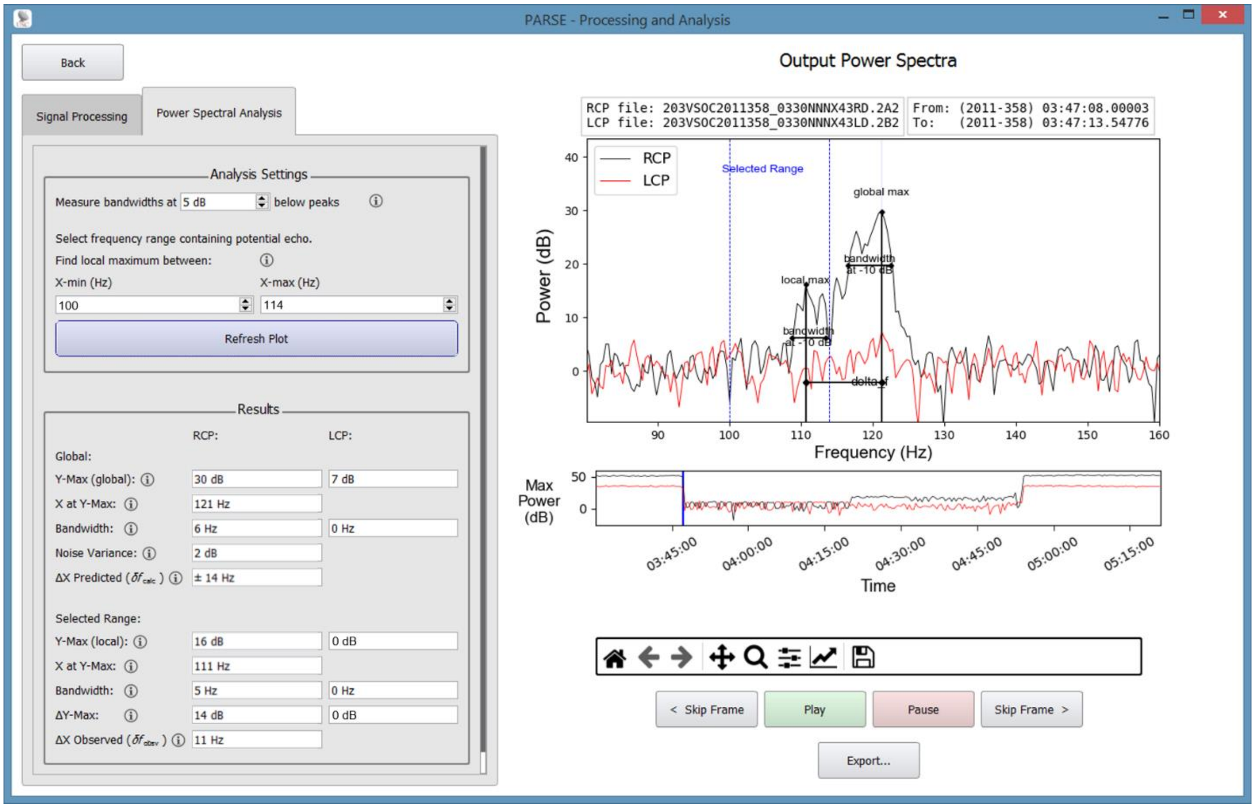


Figure 8. Final screen (power spectral analysis) of PARSE GUI. Used for surface echo detection and characterization after pausing on a specific plot of choice in the previous window.

How to Identify an Echo

Identifying features of a surface echo:

1. **Magnitude of Frequency Difference (δf) from the Direct Signal:** δf_{obsv} will be close to δf_{calc}
2. **Positive or Negative δf:** while playing the time series of power spectra, the echo will emerge on one side of the direct signal (i.e., Doppler shifted to a lower frequency due to the spacecraft moving away from Earth during entry into an occultation) and appear on the opposite side of the direct signal (a higher frequency upon exit)
3. **Echo Power:** the echo will be weaker than an un-occulted direct signal, since it is only a fraction of the power incident on the target's surface
4. **Using the Quick-View/Overview Plot:** the echo will occur during the "ramps" shown on the quick-view plot, which indicate the gradual disappearance or gradual re-emergence of the direct signal as it passes behind or re-emerges from behind the target body

5. **Validation with SPICE Geometry:** where available, NASA-produced navigational "SPICE" kernels (naif.jpl.nasa.gov) can be used to recreate the trajectory of the spacecraft and identify the time of occultation entry and occultation exit for a given orbit, hence verifying whether any "ramps" observed in the quick-view are indeed the result of an occultation. This latter step is needed because these fast changes in signal power can also be associated with orbital maneuvers or switching between different receiver modes when recording the data at the DSN.

   FOR EXAMPLE: Figure 9 – The second ramp **(e)** at 04:53:00 on the overview plot is not an exit from occultation, but rather a switch back to two-way transmitter mode. The actual occultation exit occurs at **(d)** 04:19:30 (beginning), but the 'exit' echo is not observable in this file. The exit echo was recorded separately on an "open-loop" channel by the DSN (see Asmar *et al*. 2018: Cassini Radio Science User's Guide), meaning that the 'exit' echo is contained in a separate "open-loop" RCP/LCP file (not bundled with this release).

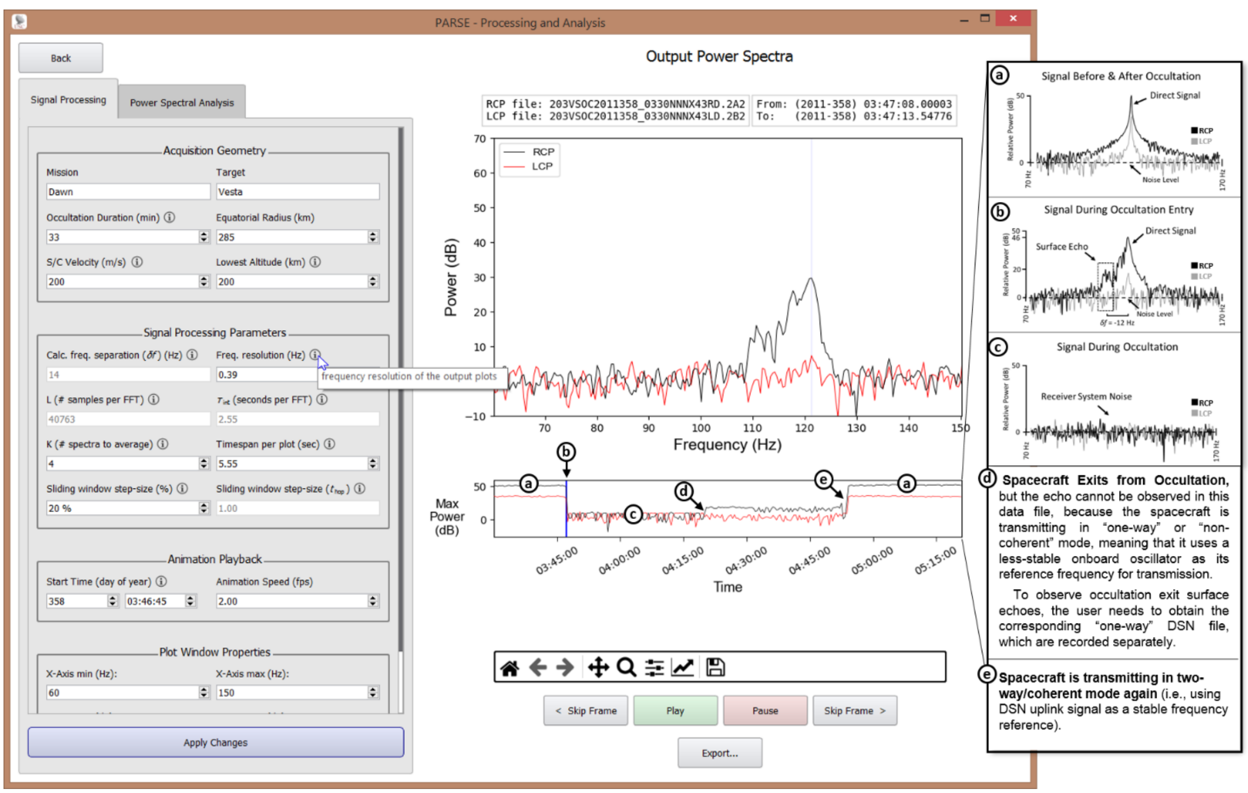


Figure 9. Annotated power spectral analysis window of the PARSE GUI. **(a)-(e)** on the quick view plot are explained on the right.

Compatible Input File Formats

Publicly available Deep Space Network radio science datasets can be found online at the NASA Planetary Data System (PDS), organized by space mission. PARSE can presently read the following specific input file formats, detailed below:

- PDS3 raw radio science files containing "I and Q" samples
- Custom/user-defined files (ASCII tables containing I and Q time series data)

PDS3: DSN Raw Radio Science Data

Raw bistatic radar or radio occultation datasets collected by the Deep Space Network's (DSN) Radio Science Receiver (RSR) are typically formatted as a binary table according to specific parameters listed in the accompanying PDS3-formatted header/label file (*.LBL). This version of PARSE is capable of handling the following type of PD3 data:

- Files must use the "detached label" sub-format of the PDS3 data standard
  - See Chapter 5.1, PDS Standards Reference (2009), version 3.8
- Once accurately stored according to the PDS3 detached label format, verify the following in the label file:

```
PDS_VERSION_ID = PDS3
RECORD_TYPE = FIXED_LENGTH
INTERCHANGE_FORMAT = BINARY
```

- Within the "TABLE" object, the radio signal data (a time series of complex-valued amplitude, called I & Q samples) must be stored in a "COLUMN" object with:

```
NAME = "SAMPLE WORDS"
DESCRIPTION = "Each ITEM contains one 32-bit sample word: quadrature (Q) sample
               data in the 16 most significant bits (MSBs) followed by in-phase
               (I) sample data in the 16 least significant bits (LSBs) [...]
```

- For an example, see the Rosetta *.LBL files that are bundled with this installation of PARSE (they are ASCII files that can be opened with a text reader)
- PDS3 mission datasets that are readable by this version of PARSE include:
  - Rosetta – Radio Science Investigation (RSI) bistatic radar data
  - Cassini – Radio Science Subsystem (RSS) bistatic and occultation data
  - Mars Express – Radio Science (MaRS) data
  - Venus Express – Venus Radio Science (VeRa) data

```
BYTES 4 THRU (M-1)
      BITS
            1 thru 8    RSR data samples; The RSR digitally samples the
received spacecraft signal with between 1-bit and 16-bit resolution.  The
samples are packed into 32-bit words.  The most significant 16 bits always
contain quadrature-phase (Q) data and the least significant 16 bits always
contain in-phase (I) data.  Between 1 to 16 samples are packed into each
32-bit word depending on how many bits per sample are used.  The time order
of the packed bits is from LSB to MSB.  Table 3-2 specifies the bit packing
for the various sample resolutions.

The RSR uses truncation to reduce the number of bits per sample to the
desired value.  This truncation creates an offset of -0.5 in the output
data stream which must be corrected in post processing software.  To
compensate for this offset each RSR data sample should be put through the
transformation 2*k + 1 where k is the 2's complement value of the 1, 2, 4,
8 or 16 bit sample. Note that the value zero is not present in this data
representation. However, all bits are used and the data is symmetric about
zero.

                   Table 3-2.  Sample Packing

==================================================================
| Bits Per Sample | Contents of 32 Bit Packed Data Register      |
==================================================================
|    16 Bits      | [Q1],[I1]                                    |
|-----------------|----------------------------------------------|
|     8 Bits      | [Q2,Q1] , [I2, I1]                           |
|-----------------|----------------------------------------------|
|     4 Bits      | [Q4, Q3, Q2, Q1], [I4, I3, I2, I1]           |
|-----------------|----------------------------------------------|
|     2 Bits      | [Q8, Q7,... Q2, Q1], [I8, I7,... I2, I1]     |
|-----------------|----------------------------------------------|
|     1 Bit       | [Q16, Q15,... Q2, Q1], [I16, I15,... I2, I1] |
==================================================================
```

Figure 10. PDS raw radio science data can be packed in different ways within a binary table (screenshot from DSN Document No. 820-013, 0159-Science). PARSE is presently compatible with 32-bit sample words containing 16-bit-resolution I and Q values as shown the top row of Table 3-2.

Dawn Mission BSR Data

Bistatic radar (BSR) data from NASA's Dawn mission was initially stored in a customized format (ASCII five-column table). However, the two specific Dawn BSR files bundled with this release of PARSE have already been converted to smaller Python-readable binary files, and are therefore accessed from the Start Screen (Figure 5) by specifically selecting "Mission: Dawn at Vesta".

Note: Due to the opportunistic nature of the BSR experiment performed by the Dawn space mission at Asteroid Vesta (Palmer et al. 2017)—in other words, planned after mission operations began—data was also provided without an accompanying PDS label files. The specific ASCII file columns included:

- Year, Day of Year, Seconds in Day, and Complex-Valued Amplitude (real and imaginary parts, i.e., the in-phase "I" and quadrature phase "Q" components)
- For the pre-bundled sample of BSR data from Dawn at Vesta, these columns have already been converted to much smaller binary Python-compatible files (*.npy)

User-Defined / Custom Dataset

Files must already be formatted as ASCII (text) files with the following characteristics:

- Each line has fixed width of 32 characters
- Each line contains 5 columns (starting positions at 0, 5, 8, 20, 26):
  - Year (YYYY)
    - 32-bit long signed integer
  - Day of Year (DOY)
    - long signed integer
  - Time of Day in Seconds (SSSSS.sssss)
    - double-precision
  - Amplitude, real part (in-phase, I)
    - long signed integer
  - Amplitude, imaginary part (quadrature phase, Q)
    - long signed integer

After clicking "User-Defined Data File" on the start screen, the user will be select one RCP file and one file containing the corresponding LCP data for simultaneous processing.

Expect PARSE to take as long as 15 minutes to process these files for the first time while it converts this large ASCII file to a compact binary format readable by Python.

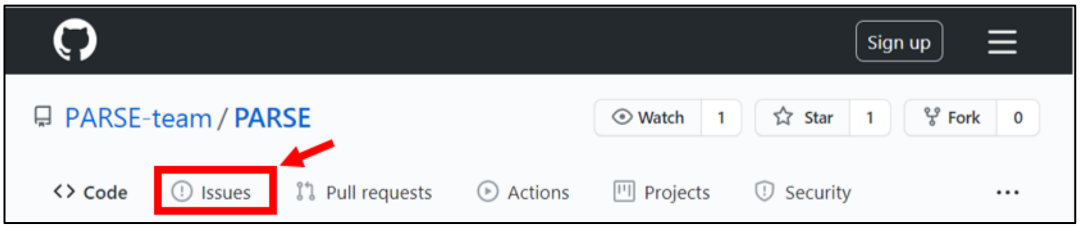
Parameter Definitions

Header in GUI	Name in GUI	Name in Python	Units	Description
Signal Processing window		`s.`		
Acquisition Geometry	Mission	`mission`		
..	Target	`target`		
..	Occultation Duration	`dt_occ`	min	typically 1 min - 30 min or longer
..	Equatorial Radius	`radius_target`	km	
..	S/C Velocity	`v_sc_orbital`	m/s	spacecraft orbital/flyby speed
..	Lowest Altitude	`altitude_sc`	km	spacecraft distance above target surface
Signal Processing Parameters	Calc. freq. separation (δf)	`df_calc`	Hz	computed frequency difference between direct and echo signals
..	Freq. resolution	`freq_res`	Hz	frequency resolution of the output plots
..	L (# samples per FFT)	`samples_per_raw_fft`		number of data points over which to perform an FFT (calculated from f_{res})
..	τ_{int} (seconds per FFT)	`seconds_per_raw_fft`	sec	FFT integration time
..	K (# spectra to average)	`raw_fft_per_average`		number of FFT's to average together
..	Timespan per plot	`seconds_for_welch`	sec	timespan of each frame/plot produced on the signal processing window
..	Sliding window step-size (%)	`percent_window_per_hop`	%	step-size between each moving average window (percentage of window size)
..	Sliding window step-size (t_{hop})	`seconds_per_hop`	sec	calculated increment between each successive moving average window
Plot Window Properties	X-Axis min	`xlim_min`	Hz	
..	X-Axis max	`xlim_max`	Hz	
..	Y-Axis min	`ylim_min`	dB	
..	Y-Axis max	`ylim_max`	dB	
Animation Playback	Start Time in File (sec)	`start_sec_user`	sec	choose from Quick View plot (time of day, sec)
..	Animation Speed	`interval`	frames/s	

Header in GUI	Name in GUI	Name in Python	Units	Description
Power Spectral Analysis window		`msmt.`		
Analysis Settings	Measure bandwidths at __ dB below peaks	`NdB_below`	dB	peak widths can be compared when measured at the same dB below their peak (e.g., comparing their "10-dB bandwidth")
.. > Select frequency range containing a potential echo	X_{min} (selected range)	`freq_local_min`	Hz	select the frequency range containing a potential echo signal but not the direct signal
.. > ..	X_{max} (selected range)	`freq_local_max`	Hz	select the frequency range containing a potential echo signal but not the direct signal
Results > Global	Y_{max} (global)	`Pxx_max_RCP`	dB	peak power in RCP (dB)
.. > ..	Y_{max} (global)	`Pxx_LCP_at_max`	dB	LCP power (dB) at the same frequency as max RCP
.. > ..	X at Y_{max}	`freq_at_max`	Hz	frequency (Hz) of the main peak
.. > ..	Bandwidth	`bandwidth_RCP_at_max`	Hz	frequency width of the main peak (measured at N-dB below the peak)
.. > ..	Bandwidth	`bandwidth_LCP_at_max`	Hz	frequency width of the LCP signal at the main RCP peak (measured at N-dB below the peak)
.. > ..	Noise Variance	`Pxx_noise_var_RCP`	dB	detectable signal peaks are defined as being at least 3 dB greater than this noise level
.. > ..	Noise Variance	`Pxx_noise_var_LCP`	dB	detectable signal peaks are defined as being at least 3 dB greater than this noise level
.. > ..	ΔX Predicted (δf_{calc})	`df_calc`	Hz	the calculated, expected frequency difference between the direct and echo peaks
Results > Selected Range	Y_{max} (local)	`Pxx_local_max_RCP`	dB	local maximum RCP power (dB) in the selected range
.. > ..	Y_{max} (local)	`Pxx_LCP_at_local_max`	dB	LCP power (dB) at the same local max
.. > ..	X at Y_{max}	`freq_at_local_max`	Hz	center frequency (Hz) of the peak in the selected range
.. > ..	Bandwidth	`bandwidth_RCP_local_max`	Hz	frequency width of the secondary RCP peak identified in the selected frequency range
.. > ..	Bandwidth	`bandwidth_LCP_at_local_max`	Hz	frequency width of the LCP signal in the selected frequency range
.. > ..	ΔY_{max}	`delta_Pxx_max_RCP`	dB	difference in peak power of the two selected maxima (dB)
.. > ..	ΔY_{max}	`delta_Pxx_LCP`	dB	difference in LCP power at the central frequencies of the two RCP peaks
.. > ..	ΔX Observed (δf_{obsv})	`df_obsv`	Hz	observed frequency difference between the two selected peaks (should be close to δf_{calc} if the two peaks are accurately identified as the direct and echo signals)

Contact Information & Bug Reporting

- For bug reporting, troubleshooting and other user feedback, please visit the official GitHub page at github.com/PARSE-team/PARSE and submit an issue (see below screenshot), or directly contact the developer (Paul Sirri) at paulsirri [at] gmail.com.



- For questions regarding the scientific background, processing algorithms or potential further applications of this tool, please contact Dr. Elizabeth M. Palmer (elizabeth.m.palmer [at] usc.edu) and/or Dr. Essam Heggy (heggy [at] usc.edu).

References & Suggested Reading

Asmar, S. W. et al. (2018). Cassini Radio Science User's Guide (version 1.1). Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.
Retrieved 8-Nov-21 from: https://pds-rings.seti.org/cassini/rss/Cassini%20Radio%20Science%20Users%20Guide%20-%2030%20Sep%202018.pdf

NASA/NAIF SPICE Website (16 Dec 2020). *SPICE: An Observation Geometry System for Space Science Missions*. NASA's Navigation and Ancillary Information Facility (NAIF). Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA.
Accessed 8-Nov-21 at: https://naif.jpl.nasa.gov/naif/

DSN 820-013, 0159-Science (Rev. B, 2008). "Radio Science Receiver Standard Formatted Data Unit (SDFU)" in *Deep Space Network External Interface Specification, JPL D-16765*. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.
Retrieved 8-Nov-21 from: https://pds-geosciences.wustl.edu/radiosciencedocs/urn-nasa-pds-radiosci_documentation/DSN_0159-SCIENCE/dsn_0159-science.2008-02-29.txt

PDS3 Standards Reference (2009, v3.8). Chapter 5: "Data Product Labels," in *Planetary Data System Standards Reference, JPL D-7669, Part 2*. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.
Retrieved 8-Nov-21 from: https://pds.nasa.gov/datastandards/pds3/standards/

Palmer, E. M., Heggy, E. & Kofman, W. (2017). Orbital bistatic radar observations of asteroid Vesta by the Dawn mission. *Nature Communications*, 8(409), 1–12. doi:10.1038/s41467-017-00434-6

Palmer, E. M. & Heggy, E. (2020). Bistatic Radar Occultations of Planetary Surfaces. *IEEE Geoscience Remote Sensing Letters*, 17(5), 804–808. doi:10.1109/LGRS.2019.2931310

Simpson, R. A. (1993). Spacecraft studies of planetary surfaces using bistatic radar. IEEE Transactions on Geoscience and Remote Sensing, 31(2), 465–482. doi:10.1109/36.214923

Simpson, R. A., Tyler, G. L., Pätzold, M., Häusler, B., Asmar, S. W. & Sultan-Salem, A. K. (2011). Polarization in Bistatic Radar Probing of Planetary Surfaces: Application to Mars Express Data. *Proceedings of the IEEE*, 99(5), 858–874. doi:10.1109/JPROC.2011.2106190

Simpson, R. & Tyler, G. L. (2001). Mars Global Surveyor Bistatic Radar Probing of the MPL/DS2 Target Area. *Icarus*, 152(1), 70–74. doi:10.1006/icar.2001.6629