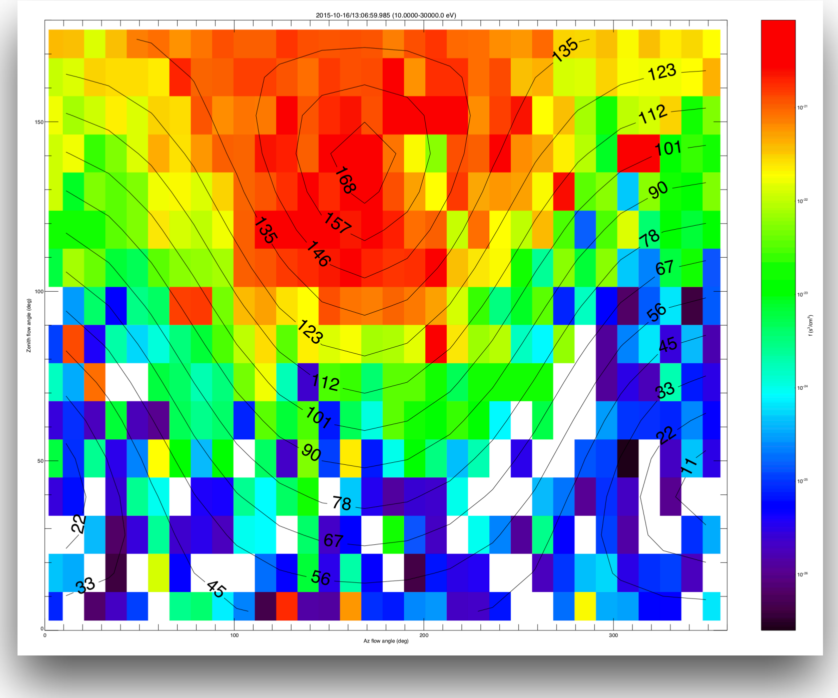
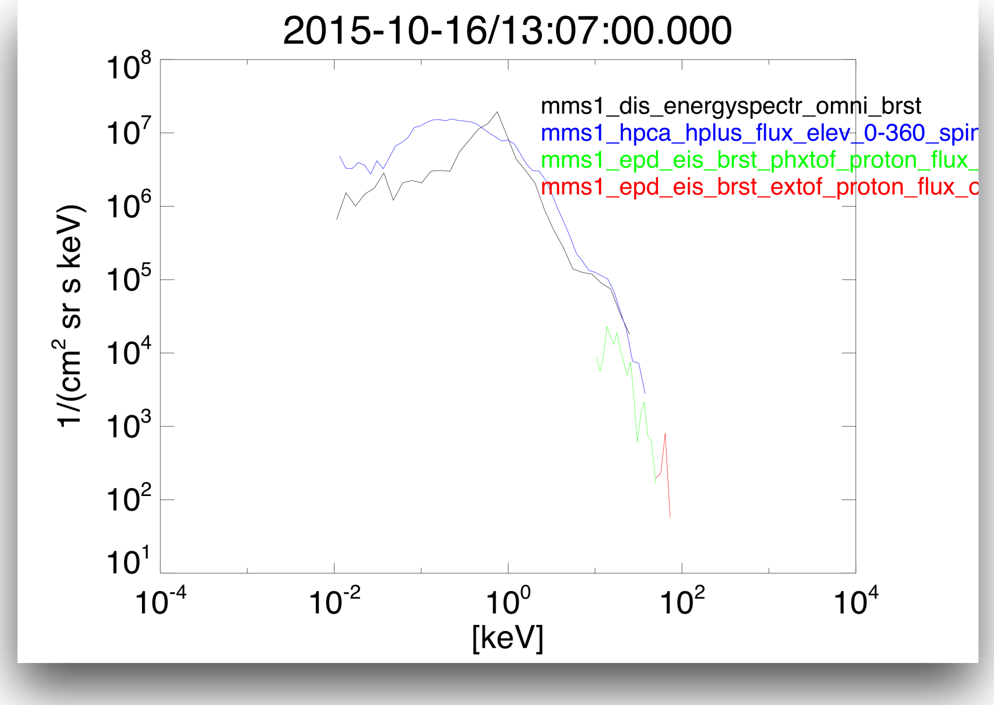
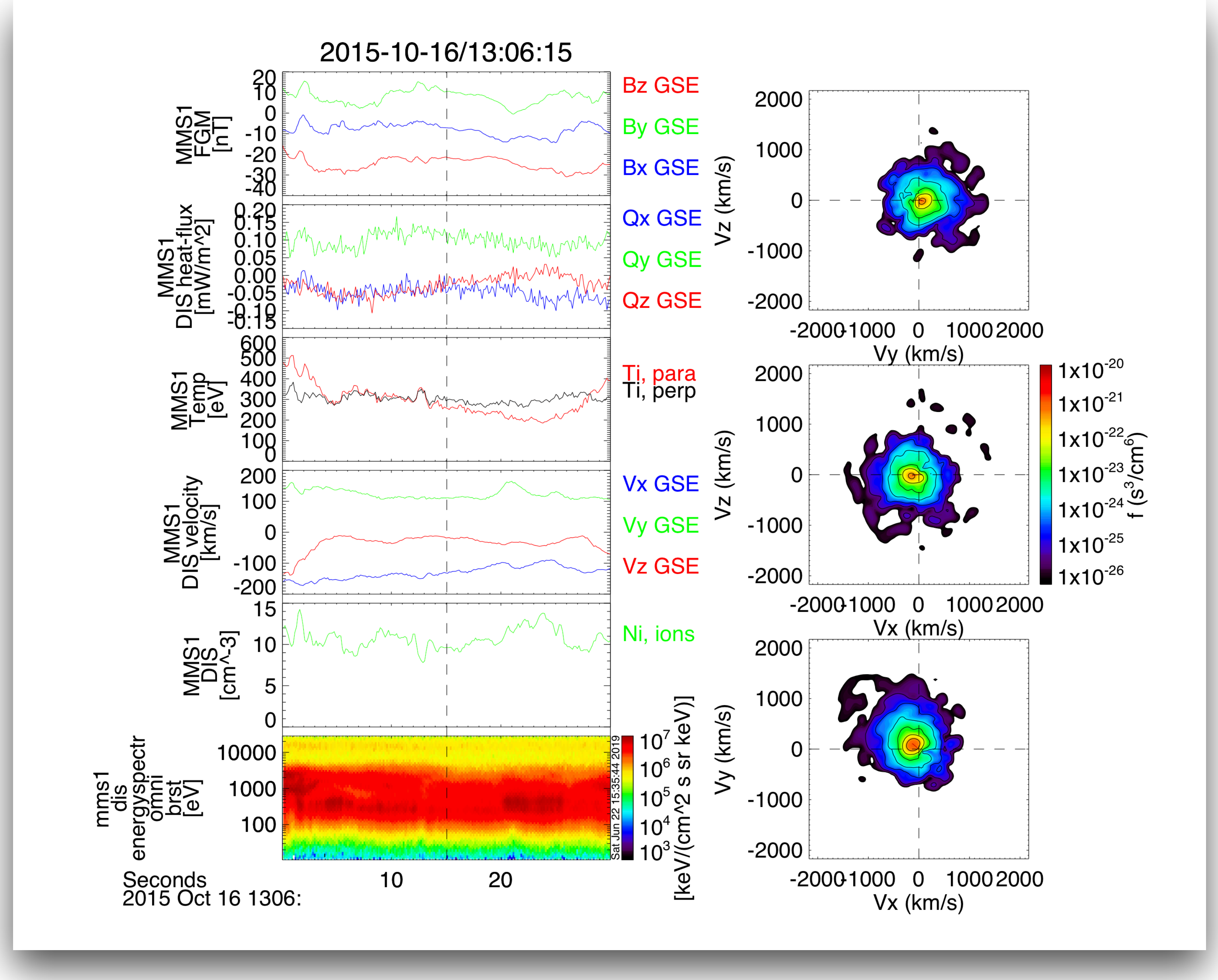






Analyzing MMS Data with SPEDAS (and pySPEDAS)

Eric Grimes (egrimes@igpp.ucla.edu)

June 27, 2019

Overview

What's New / Plug-in Status

IDL

- Getting Started
- Loading and Plotting Data
- Analysis Tools

Python

- Getting Started
- Loading and Plotting Data
- Current Status

What's New / Plug-in Status

- Recently began developing MMS plug-in for pySPEDAS
- Several improvements to IDL plug-in, including:
  - MMS event search
  - flatten_spectra unit conversions
  - New routine for removing old MMS data files
  - 500+ unit/regression tests, all passing

Getting Started

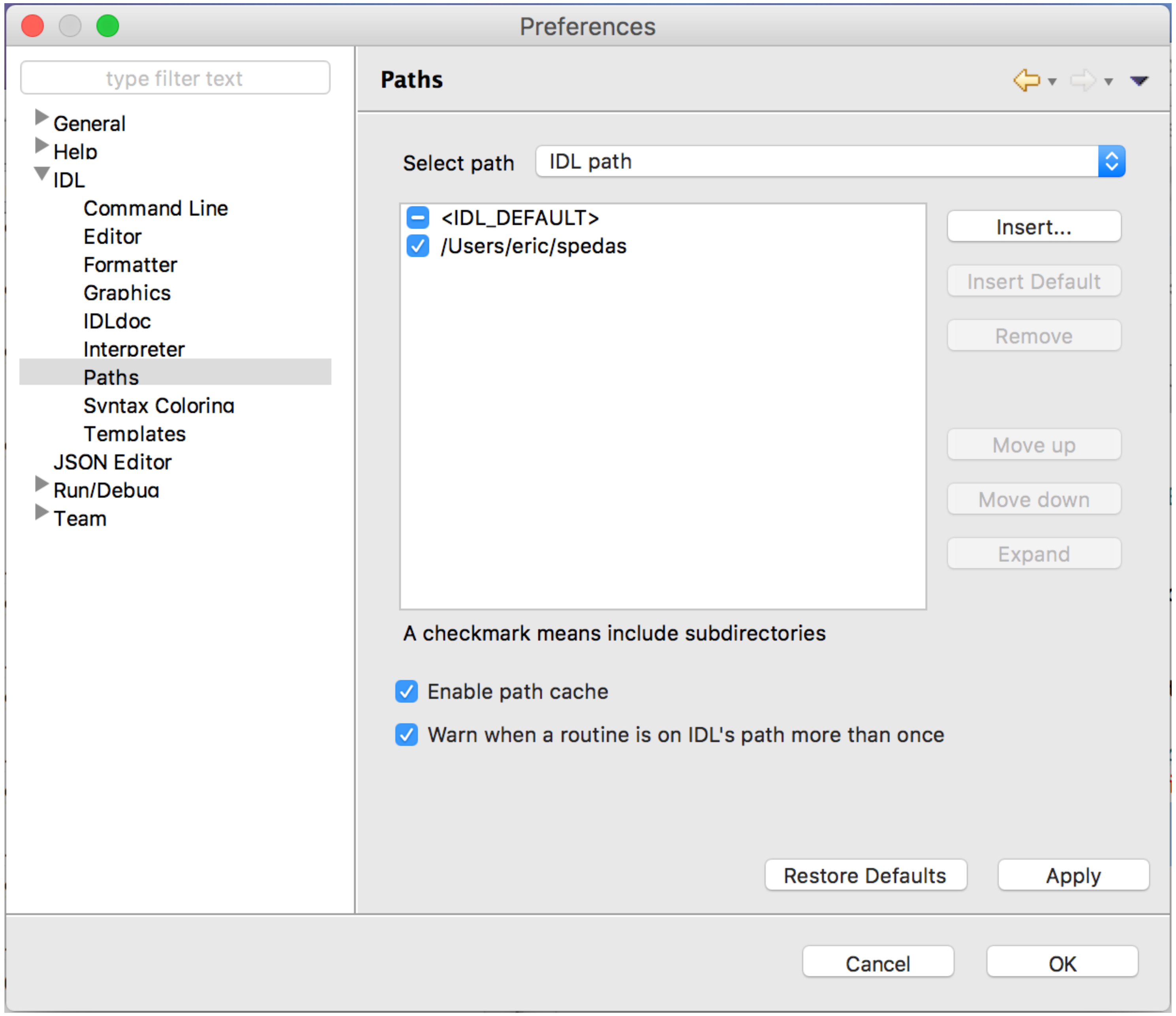
Requirements

- Windows, Linux, or OS X
- IDL 8.4+
- IDL CDF Library 3.6.3+

Getting Started

Installing SPEDAS

http://spedas.org/wiki/index.php?title=Downloads_and_Installation



Select path: IDL path

- <IDL_DEFAULT>
- ☑ /Users/eric/spedas

A checkmark means include subdirectories

☑ Enable path cache

☑ Warn when a routine is on IDL's path more than once

Getting Started

Organization

spedas_gui: Components of the SPEDAS Graphical User Interface (GUI)

external: Code developed externally, but distributed with SPEDAS (CDAWeb, Geopack, etc.)

projects: Mission specific code

general: General science analysis tools

Getting Started

MMS Load Routines

mms_load_fgm	Fluxgate Magnetometer
mms_load_scm	Search-coil Magnetometer
mms_load_fsm	L3 FGM+SCM
mms_load_mec	Ephemeris and Coordinates
mms_load_fpi	Fast Plasma Investigation
mms_load_hpca	Hot Plasma Composition Analyzer
mms_load_eis	Energetic Ion Spectrometer
mms_load_feeps	Fly’s Eye Energetic Particle Sensor
mms_load_edp	Electric-field Double Probe
mms_load_edi	Electron Drift Instrument
mms_load_dsp	Digital Signal Processor
mms_load_aspoc	Active Spacecraft Potential Control
mms_load_tetrahedron_qf	Tetrahedron Quality Factor
mms_load_fast_segments	Fast survey intervals
mms_load_brst_segments	Burst intervals

Getting Started

Standard Keywords

• trange	`trange=['2015-10-16', '2015-10-17']`
• probes	`probes=[1, 2, 3, 4]`
• level	`level='l2'`
• data_rate	`data_rate='srvy'`
• datatype	`datatype=['des-moms', 'dis-moms']`

Getting Started

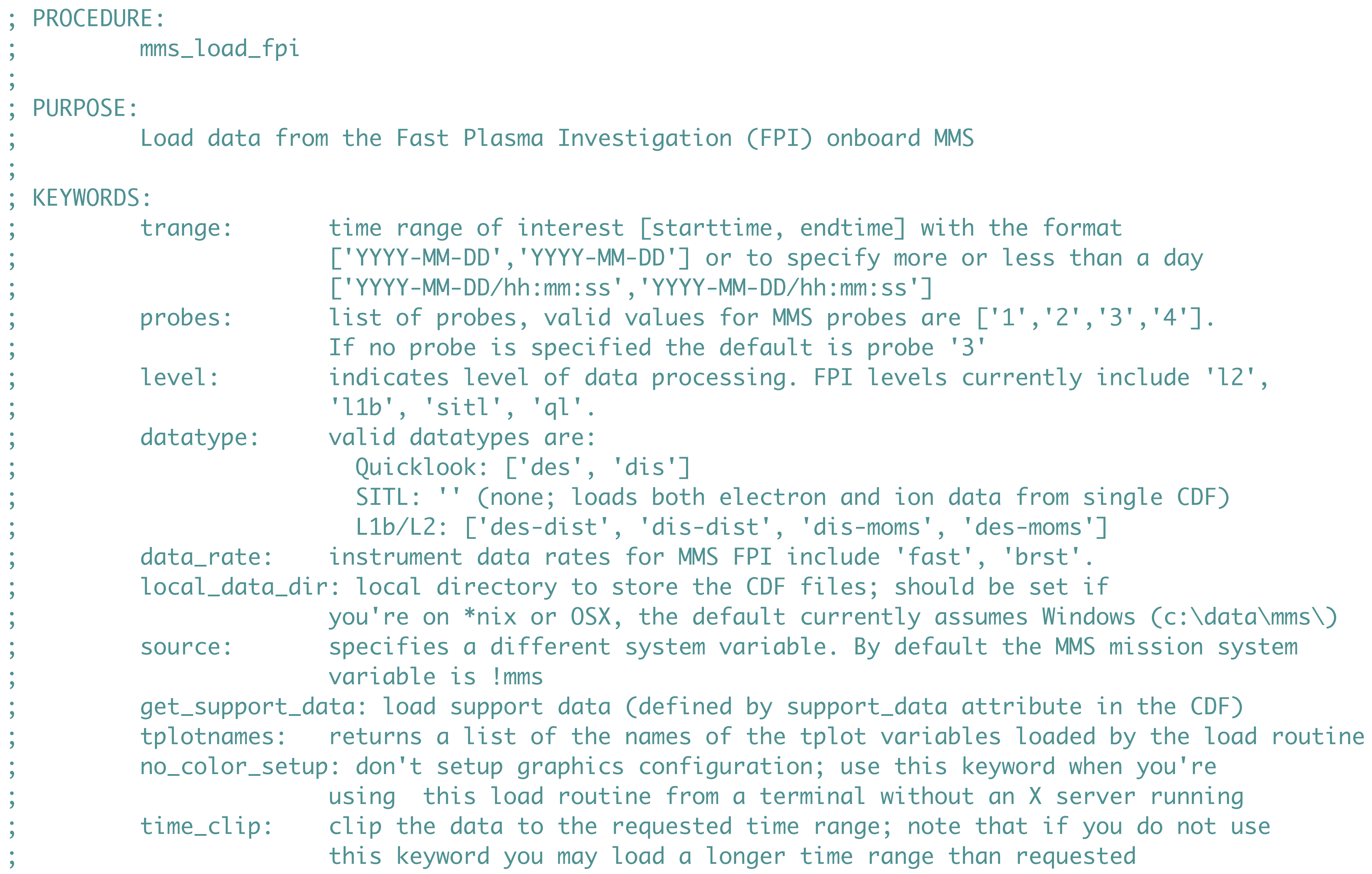
Standard Keywords

• suffix	`suffix='_burst_mode'`
• time_clip	`/time_clip`
• no_update	`/no_update`
• spdf	`/spdf`
• tplotnames	`tplotnames=tvarnames`

Getting Started

Standard Keywords

For a complete list of keywords and their descriptions, see the header for the load routine you’re interested in, e.g.,



```
; PROCEDURE:
;         mms_load_fpi
;
; PURPOSE:
;         Load data from the Fast Plasma Investigation (FPI) onboard MMS
;
; KEYWORDS:
;         trange:       time range of interest [starttime, endtime] with the format
;                       ['YYYY-MM-DD','YYYY-MM-DD'] or to specify more or less than a day
;                       ['YYYY-MM-DD/hh:mm:ss','YYYY-MM-DD/hh:mm:ss']
;         probes:       list of probes, valid values for MMS probes are ['1','2','3','4'].
;                       If no probe is specified the default is probe '3'
;         level:        indicates level of data processing. FPI levels currently include 'l2',
;                       'l1b', 'sitl', 'ql'.
;         datatype:     valid datatypes are:
;                         Quicklook: ['des', 'dis']
;                         SITL: '' (none; loads both electron and ion data from single CDF)
;                         L1b/L2: ['des-dist', 'dis-dist', 'dis-moms', 'des-moms']
;         data_rate:    instrument data rates for MMS FPI include 'fast', 'brst'.
;         local_data_dir: local directory to store the CDF files; should be set if
;                       you're on *nix or OSX, the default currently assumes Windows (c:\data\mms\)
;         source:       specifies a different system variable. By default the MMS mission system
;                       variable is !mms
;         get_support_data: load support data (defined by support_data attribute in the CDF)
;         tplotnames:   returns a list of the names of the tplot variables loaded by the load routine
;         no_color_setup: don't setup graphics configuration; use this keyword when you're
;                       using  this load routine from a terminal without an X server running
;         time_clip:    clip the data to the requested time range; note that if you do not use
;                       this keyword you may load a longer time range than requested
```

Loading and Plotting Data

Simple Example

```
; load the MMS1 FGM data for October 16, 2015
mms_load_fgm, probe=1, data_rate='srvy', trange=['2015-10-16', '2015-10-17']

; list the tplot variables loaded
tplot_names

; get the data out of a tplot variable
get_data, 'mms1_fgm_b_gsm_srvy_l2_bvec', data=bgsm_vec, dlimits=bgsm_metadata

; get basic info on the IDL vars that hold the B-field data
help, bgsm_vec ; structure, x: times, y: data (x, y, z)
help, bgsm_metadata ; plotting and file metadata

; store the data into a different tplot variable
store_data, 'new_var_with_b_gsm', data=bgsm_vec, dlimits=bgsm_metadata

; plot the newly created variable
tplot, 'new_var_with_b_gsm'
```

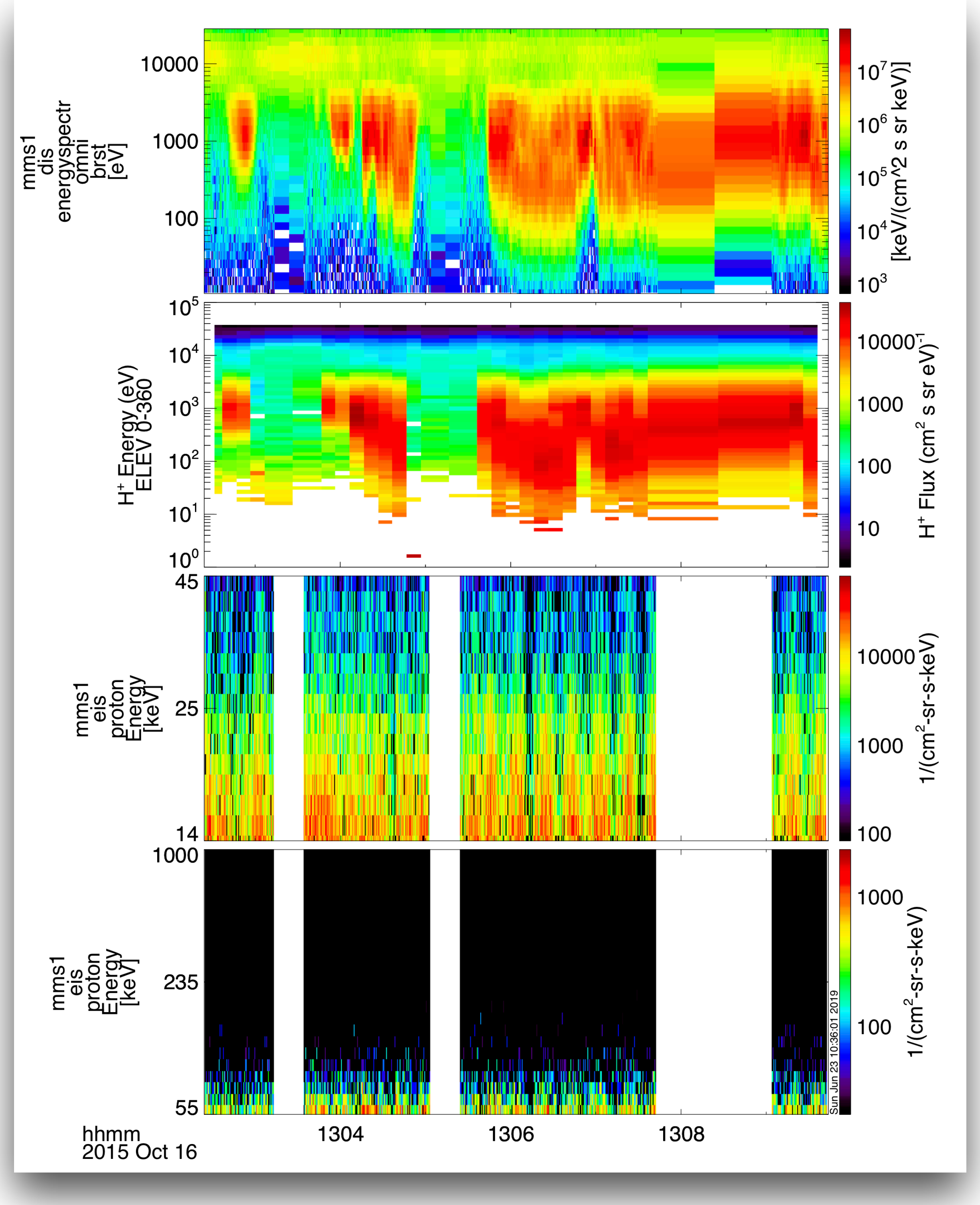
Loading and Plotting Data

Simple Example - Flux

```
trange = ['2015-10-16/13', '2015-10-16/13:10']

mms_load_fpi, probe=1, data_rate='brst', datatype=['dis-moms'], trange=trange, /time_clip

mms_load_eis, datatype=['extof', 'phxtof'], data_rate='brst', trange=trange, /time_clip, probe=1

mms_load_hpca, data_rate='brst', trange=trange, /time_clip, probe=1, datatype='ion', /major
mms_hpca_calc_anodes, fov=[0, 360]
mms_hpca_spin_sum, probe=1, /avg

tplot, ['mms1_dis_energyspectr_omni_brst', $
  'mms1_hpca_hplus_flux_elev_0-360_spin', $
  'mms1_epd_eis_brst_phxtof_proton_flux_omni', $
  'mms1_epd_eis_brst_extof_proton_flux_omni']
```

Loading and Plotting Data

Simple Example - Flux



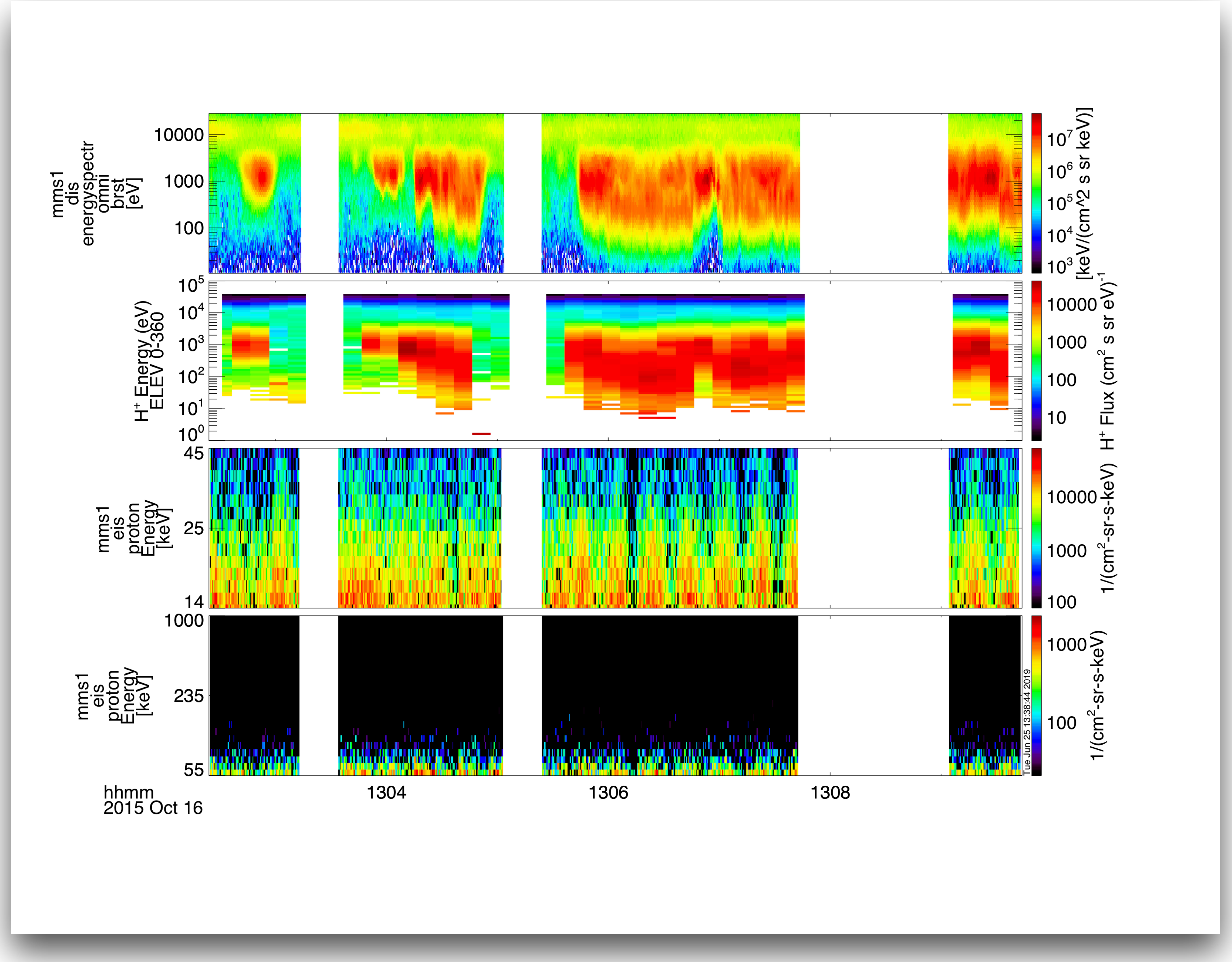
Loading and Plotting Data

De-gapping

```
; remove interpolation in the gaps
tdegap, ['mms1_dis_energyspectr_omni_brst', $
'mms1_hpca_hplus_flux_elev_0-360_spin', $
  'mms1_epd_eis_brst_phxtof_proton_flux_omni', $
  'mms1_epd_eis_brst_extof_proton_flux_omni'], /overwrite

tplot, ['mms1_dis_energyspectr_omni_brst', $
    'mms1_hpca_hplus_flux_elev_0-360_spin', $
    'mms1_epd_eis_brst_phxtof_proton_flux_omni', $
    'mms1_epd_eis_brst_extof_proton_flux_omni']
```

Loading and Plotting Data

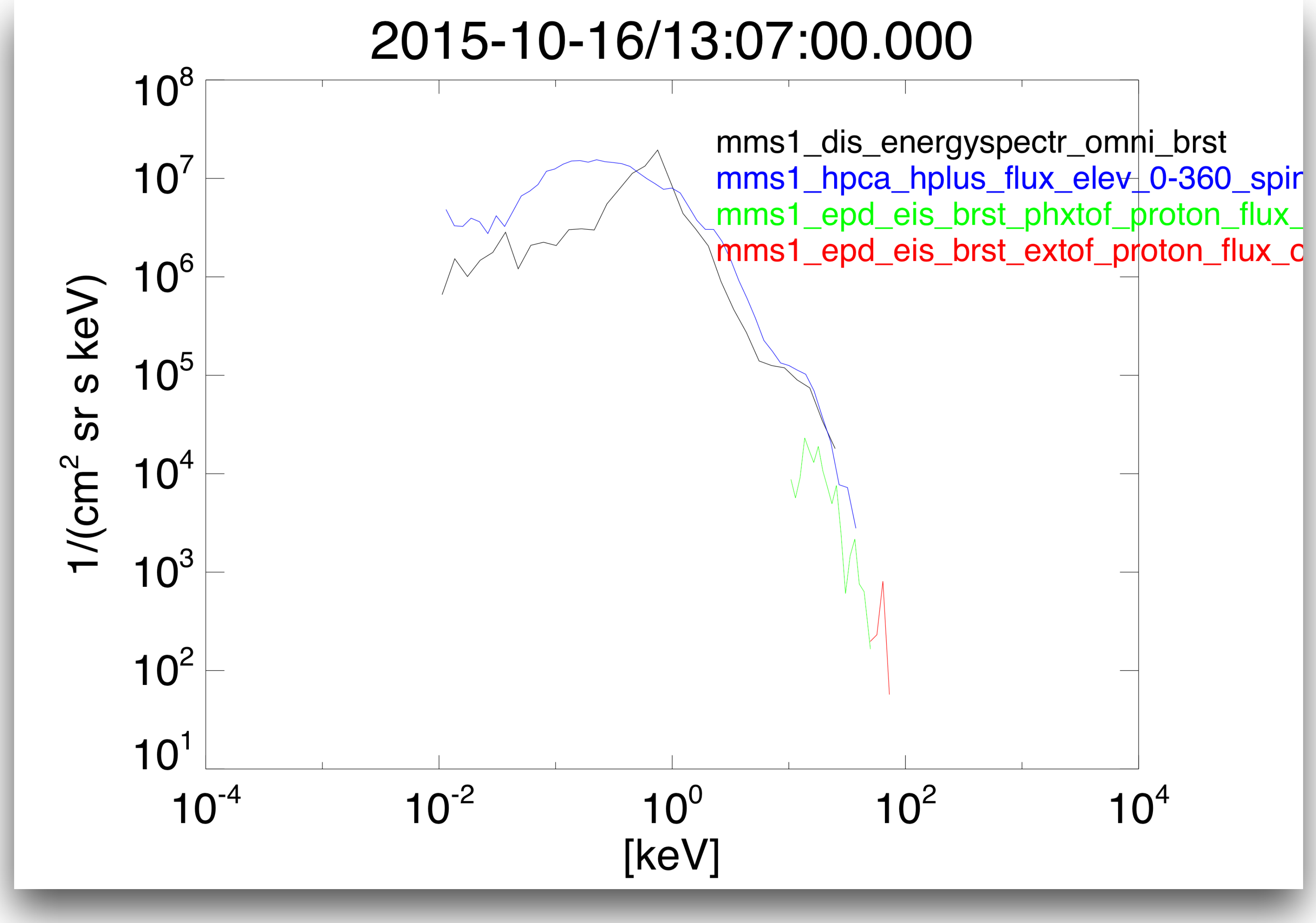
De-gapping



Loading and Plotting Data

Creating line plots at a time

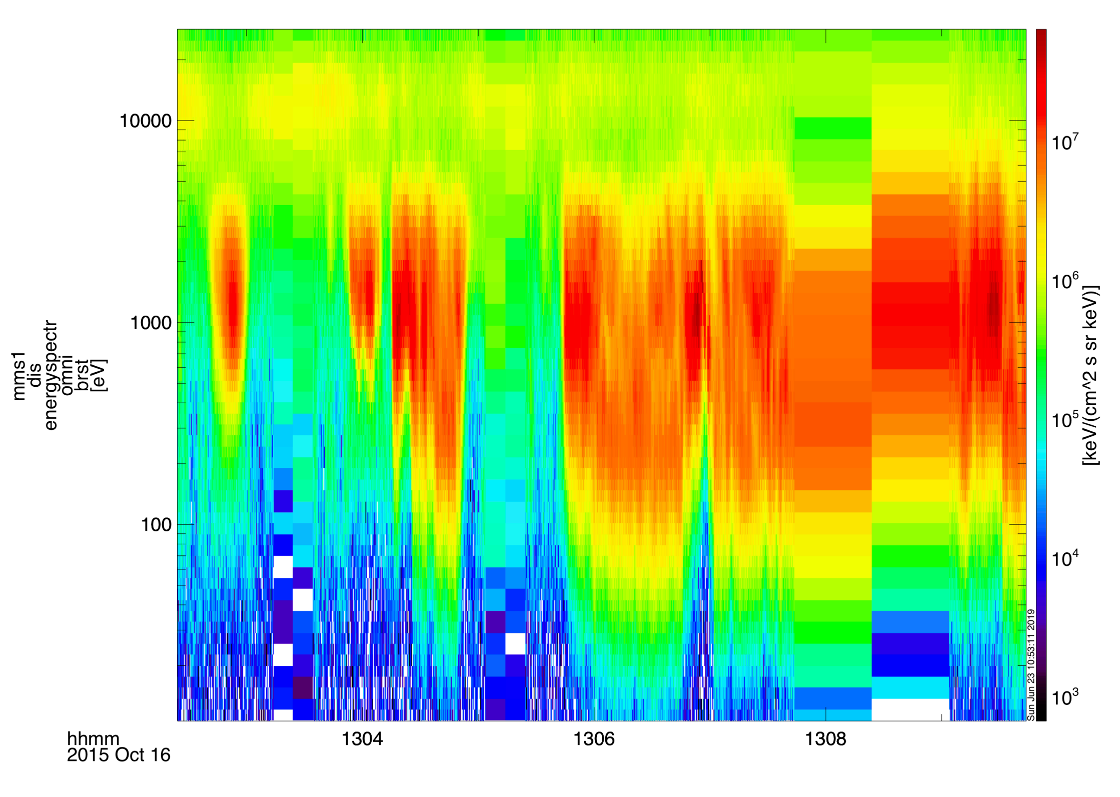
```
flatten_spectra, /to_flux, /to_kev, /xlog, /ylog, time='2015-10-16/13:07'
```

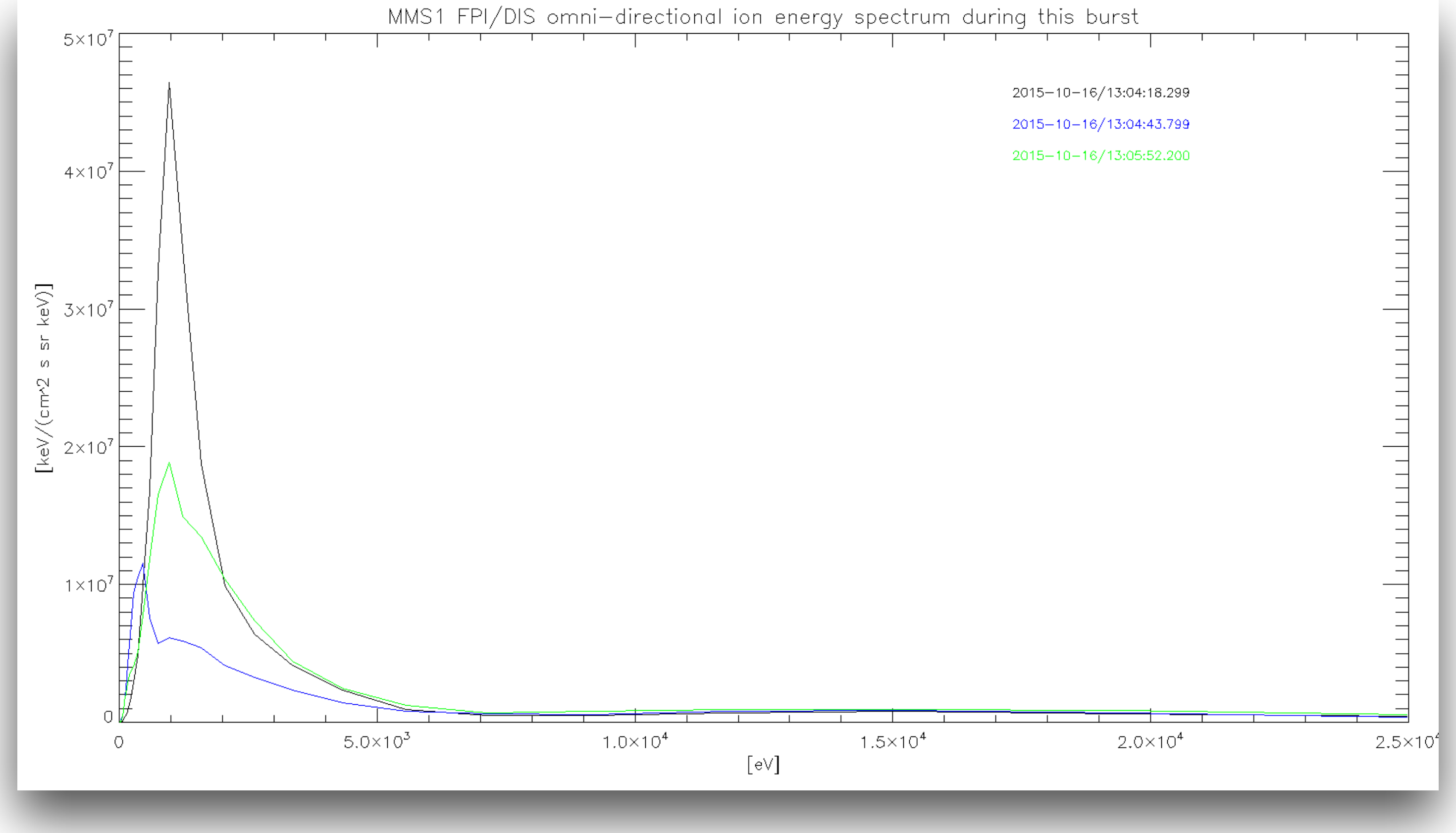


Loading and Plotting Data

Creating line plots at a time

```
tplot, 'mms1_dis_energyspectr_omni_brst'

flatten_spectra_multi, 3
```



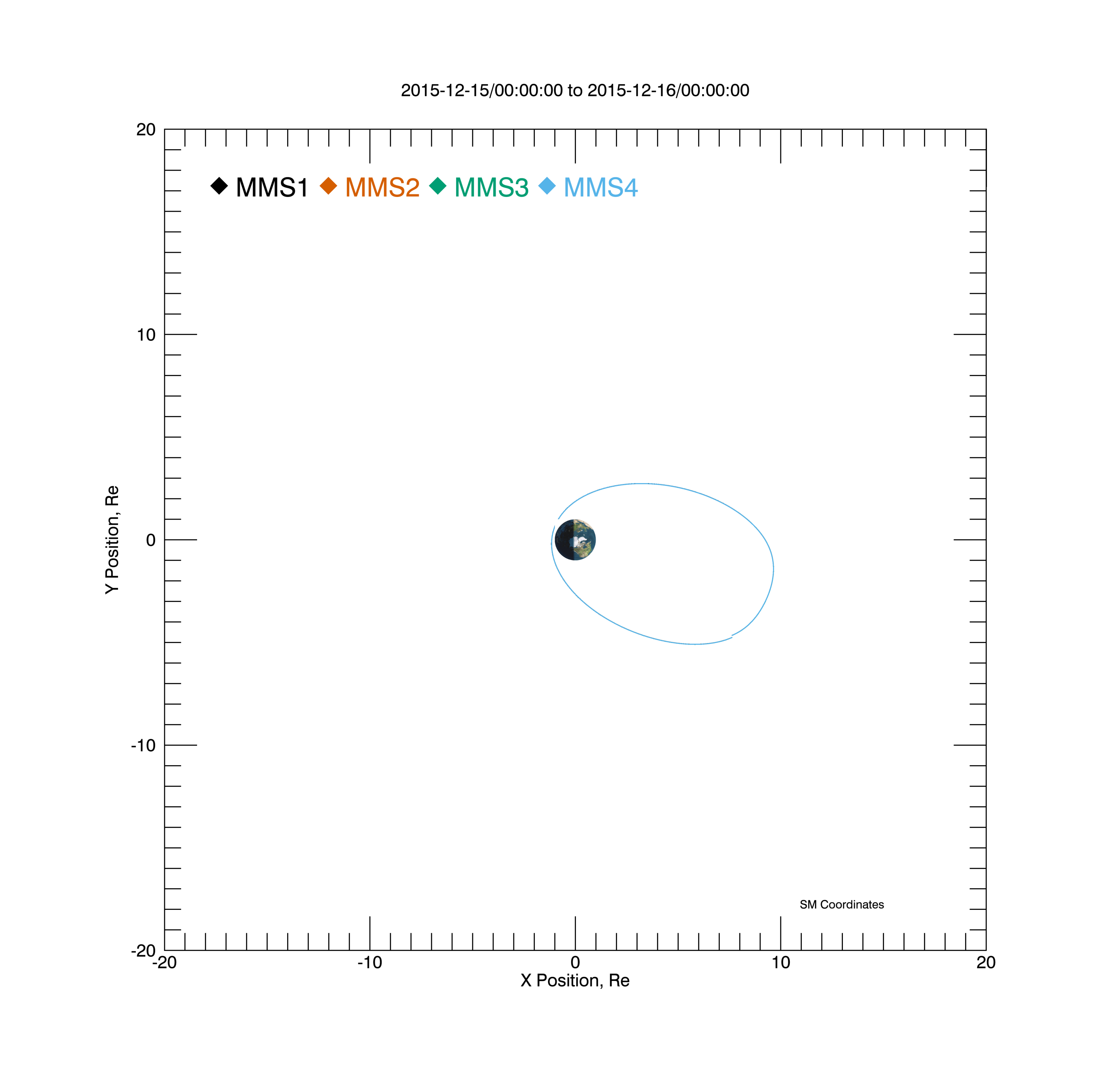


What's New / Plug-in Status
IDL
Getting Started
Loading and Plotting Data
Analysis Tools
Python
Getting Started
Loading and Plotting Data
Current Status

Loading and Plotting Data

Plotting Orbits

```
mms_orbit_plot, coord='sm', probe=[1, 2, 3, 4], trange=['2015-12-15', '2015-12-16'], yrange=[-20, 20], xrange=[-20, 20]
```



Loading and Plotting Data

MMS event search

```
mms_event_search, 'bz', authors=authors, descriptions=descriptions, start_times=start_times, end_times=end_times

print, 'Description: ' + descriptions[0]
print, 'Author: ' + authors[0]
print, 'Start time: ' + time_string(start_times[0])
print, 'End time: ' + time_string(end_times[0])
```

```
Description: Sharp fluctuations in Bz
Author: fwilder(EVA)
Start time: 2015-07-23/19:45:44
End time: 2015-07-23/19:53:44
```

Loading and Plotting Data

Remove old data files

To remove old MMS data files stored on your machine, use:

```
mms_remove_old_files
```

Analysis Tools

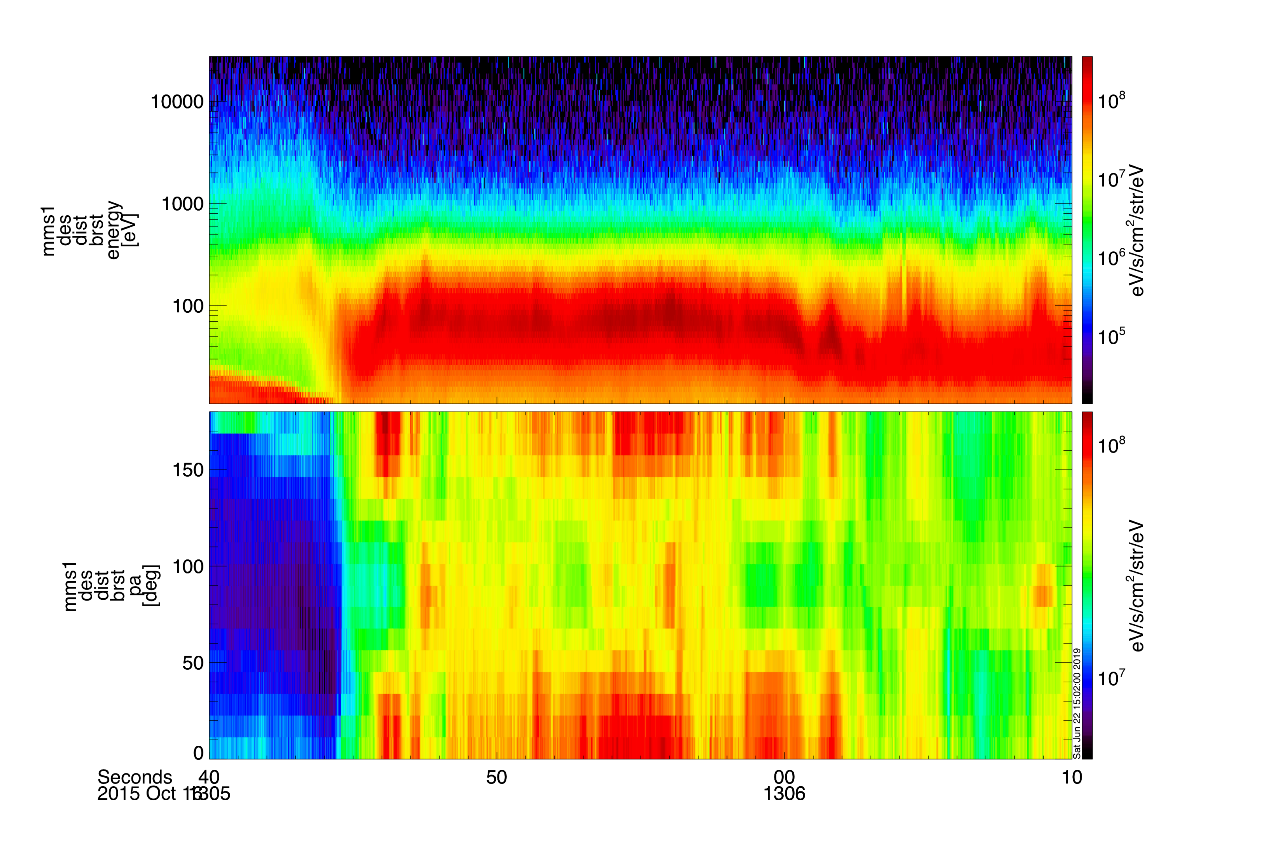
mms_part_getspec	Calculate spectra from FPI/HPCA distributions
mms_part_slice2d	Plot 2D slices of FPI/HPCA distributions
mms_part_isee3d	Plot FPI/HPCA distributions in 3D
mms_flipbookify	Combine FPI/HPCA 2D slices with tplot windows
eis_ang_ang	Create EIS angle-angle plots
mms_fpi_ang_ang	Create FPI angle-angle plots
mms_curl	Curlometer technique

What's New / Plug-in Status
IDL
    Getting Started
    Loading and Plotting Data
    Analysis Tools
Python
    Getting Started
    Loading and Plotting Data
    Current Status

Analysis Tools

Calculate spectra from FPI/HPCA distributions

```
; use short time range for data due to high resolution
timespan, '2015-10-16/13:05:40', 30, /sec

; generate products
mms_part_getspec, instrument='fpi', probe='1', species='e', data_rate='brst', level='l2', outputs=['energy', 'pa']

; plot spectrograms
tplot, 'mms1_des_dist_brst_'+['energy', 'pa']
```

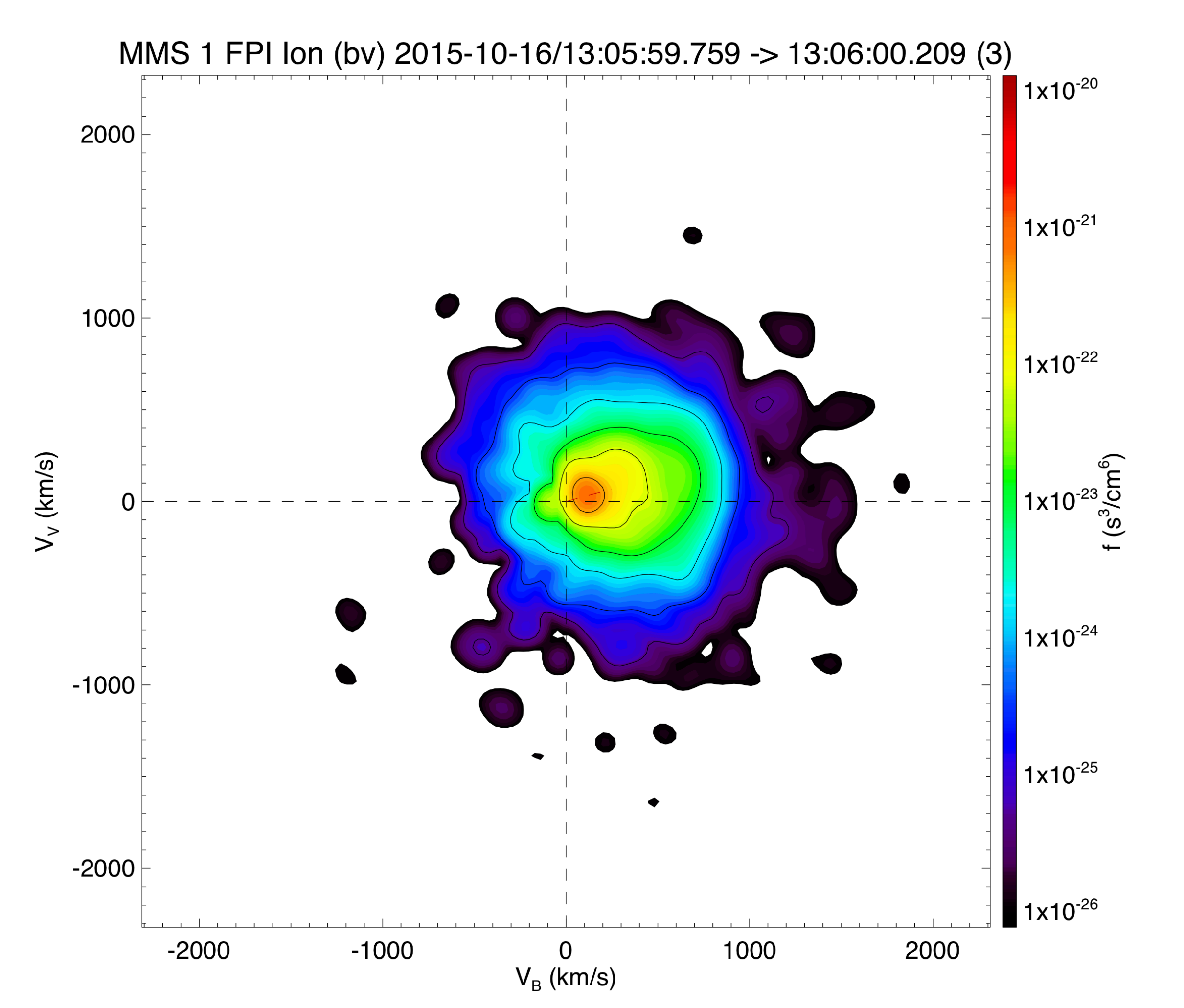


What's New / Plug-in Status
IDL
    Getting Started
    Loading and Plotting Data
    Analysis Tools
Python
    Getting Started
    Loading and Plotting Data
    Current Status

Analysis Tools

Plot 2D slices of FPI/HPCA distributions

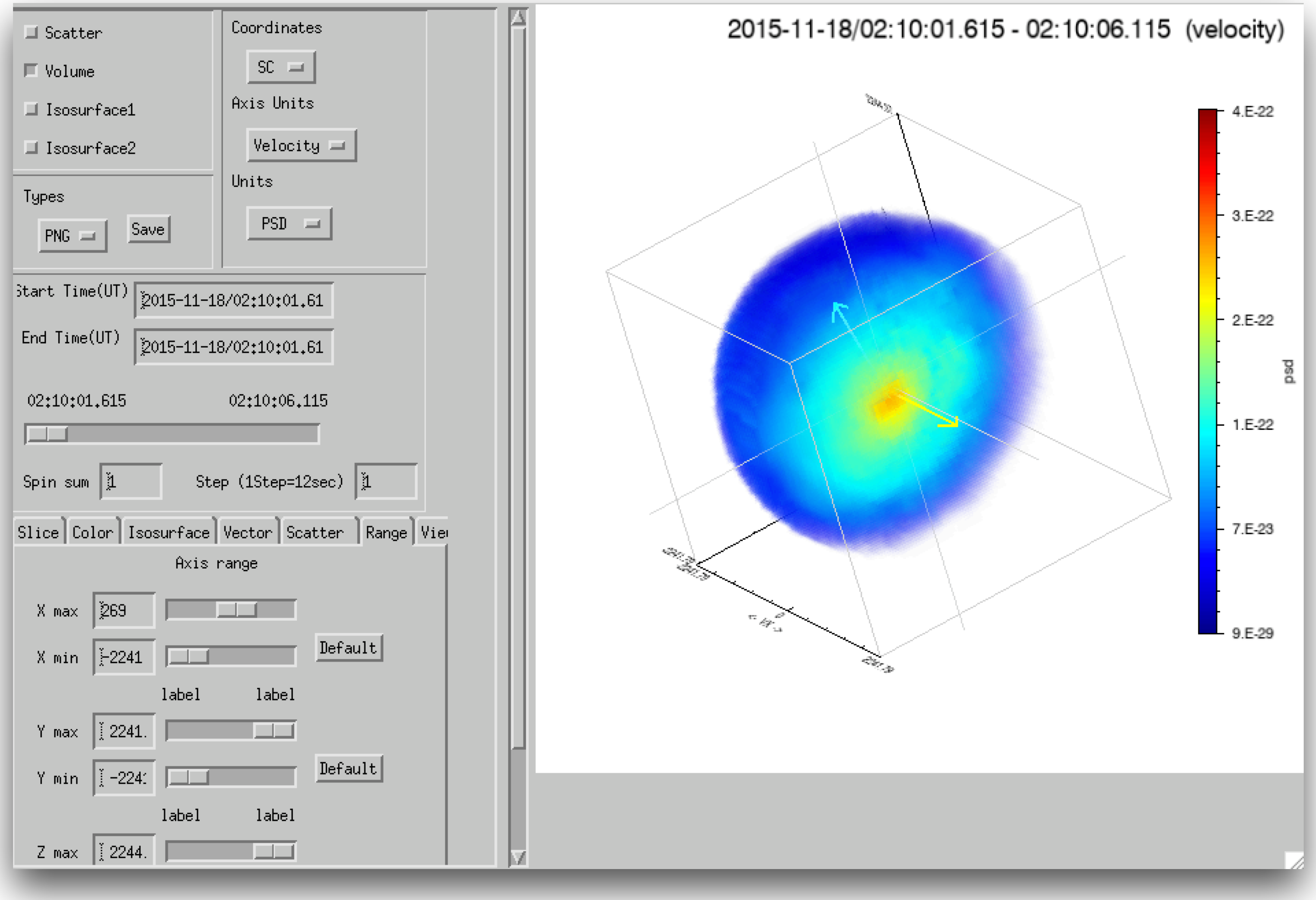
```
;field/velocity aligned slice
;  -the plot's x axis is parallel to the B field
;  -the plot's y axis is defined by the bulk velocity direction
;---------------------------------------------
mms_part_slice2d, rotation='bv', samples=3, time='2015-10-16/13:06:00', probe=1, species='i', data_rate='brst'
```



Analysis Tools

Plot FPI/HPCA distributions in 3D

```
timespan, '2015-11-18/02:10:00', 10, /sec

mms_part_isee3d, probe='1', species='i', data_rate='fast', level='l2'
```



☐ Scatter
☑ Volume
☐ Isosurface1
☐ Isosurface2

Coordinates: SC
Axis Units: Velocity
Units: PSD

Types: PNG | Save

Start Time(UT): 2015-11-18/02:10:01.61
End Time(UT): 2015-11-18/02:10:01.61

02:10:01.615 02:10:06.115

Spin sum: 1
Step (1Step=12sec): 1

Axis range

X max: 269
X min: -2241
Y max: 2241.
Y min: -224:
Z max: 2244.

2015-11-18/02:10:01.615 - 02:10:06.115 (velocity)

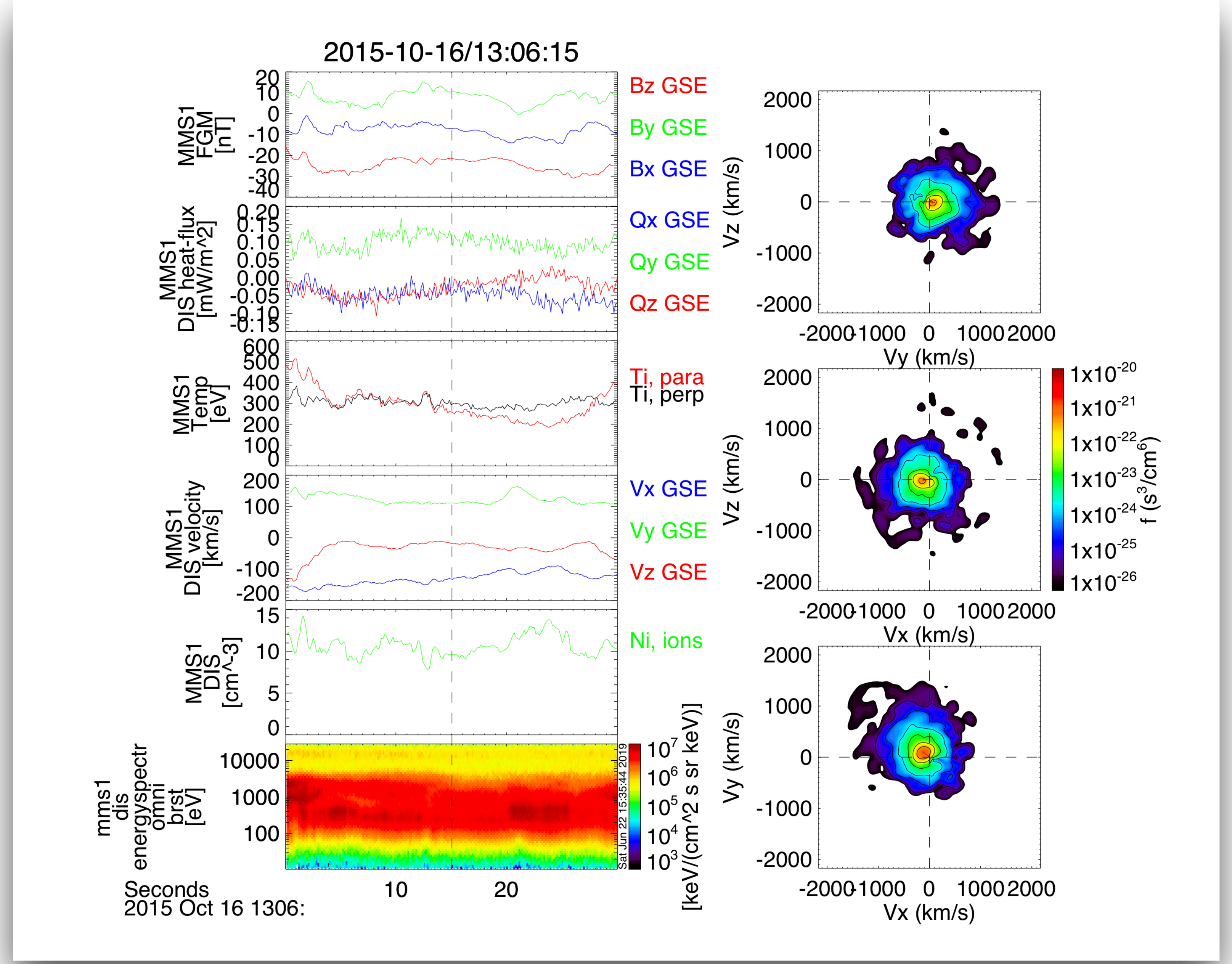
Analysis Tools

Combine FPI/HPCA 2D slices with tplot windows

```
trange=['2015-10-16/13:06:00', '2015-10-16/13:06:30']
probe=1
data_rate = 'brst'
species = 'i'

mms_load_fgm, trange=trange, probe=probe, /time_clip
mms_load_fpi, trange=trange, probe=probe, datatype='d'+species+'s-moms', /time_clip, data_rate=data_rate

window, xsize=1000, ysize=650

; store the temperature in the same panel
store_data, 'temp', data='mms1_d'+species+'s_temppara_brst mms1_d'+species+'s_tempperp_brst'

tplot, ['mms1_fgm_b_gse_srvy_l2_bvec', 'mms1_dis_heatq_gse_brst', 'temp', 'mms1_d'+species+'s_bulkv_gse_brst', $
  'mms1_d'+species+'s_numberdensity_brst', 'mms1_d'+species+'s_energyspectr_omni_brst']

mms_flipbookify, time_step=10, probe=1, species='i', /postscript
```

Analysis Tools

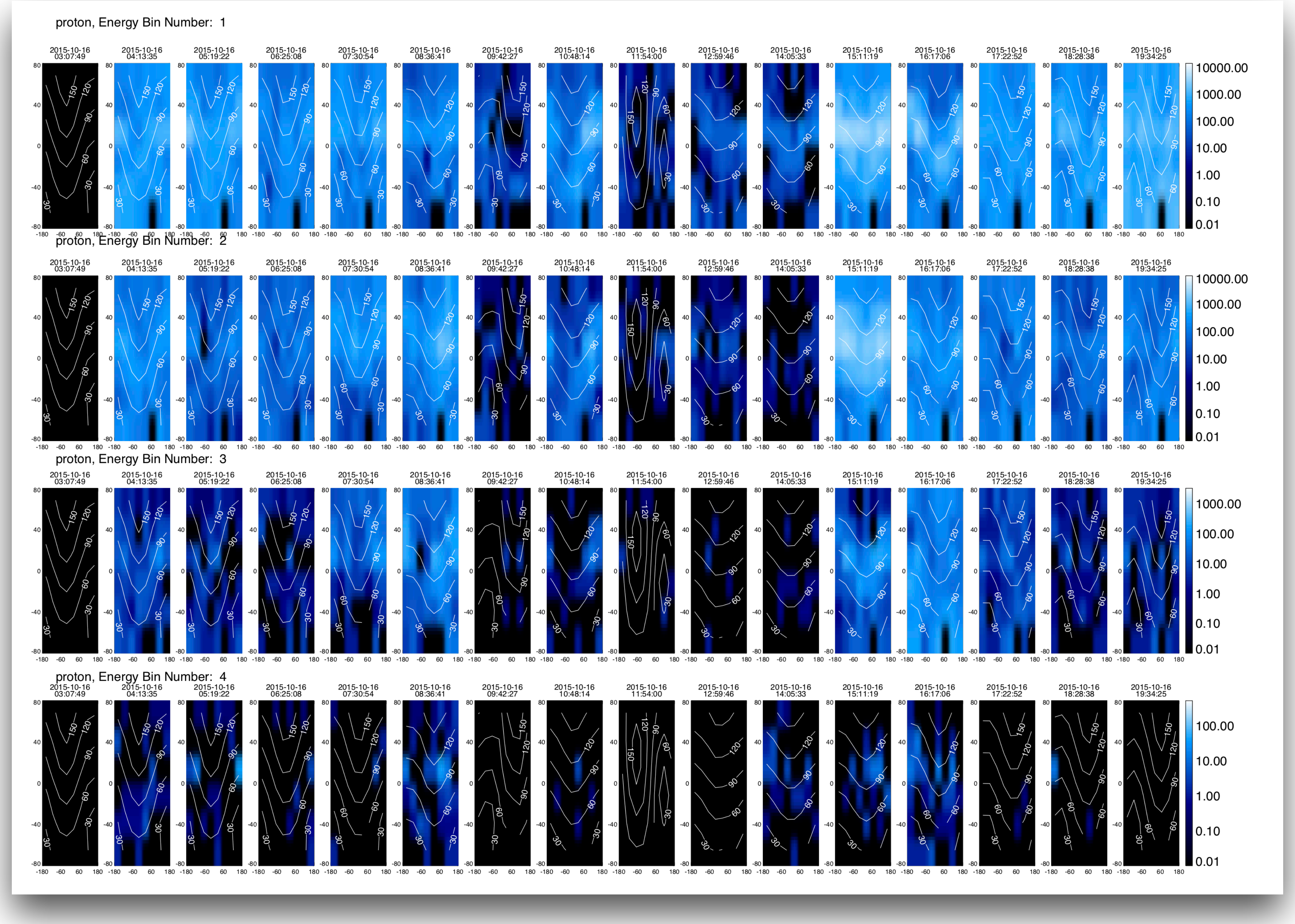
Combine FPI/HPCA 2D slices with tplot windows



Analysis Tools

Create EIS angle-angle plots

```
eis_ang_ang, trange=['2015-10-16', '2015-10-17'], level='l2', probe=3, datatype='extof'
```



proton, Energy Bin Number: 1

proton, Energy Bin Number: 2

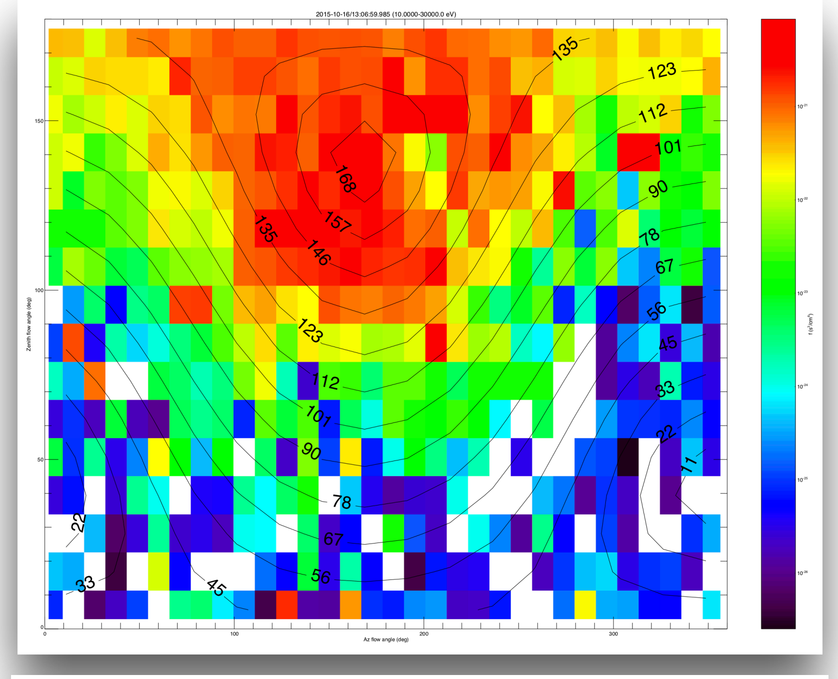
proton, Energy Bin Number: 3

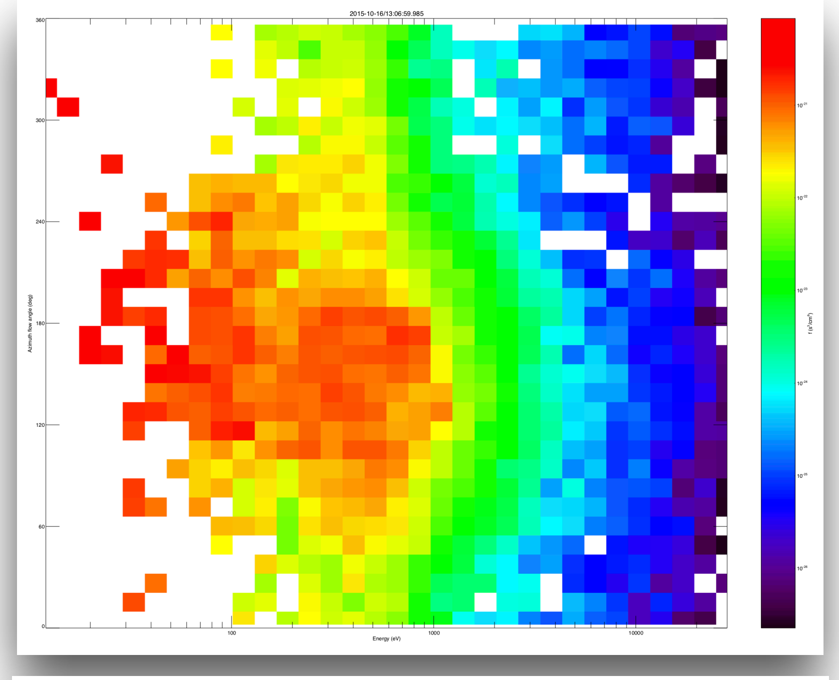
proton, Energy Bin Number: 4

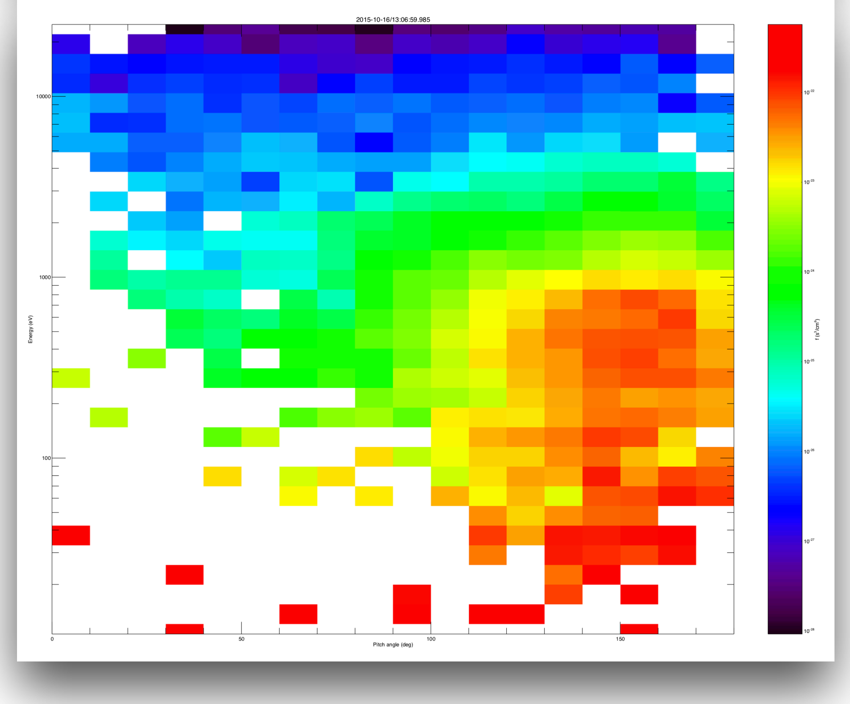
Analysis Tools

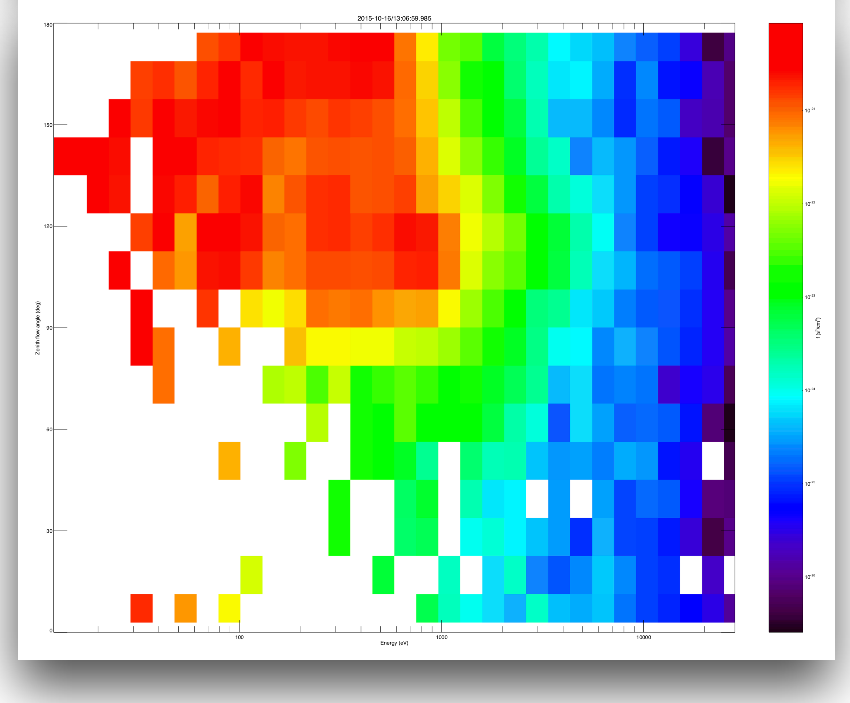
Create FPI angle-angle plots

```
mms_fpi_ang_ang, '2015-10-16/13:06:59.985', species='i', data_rate='brst'
```





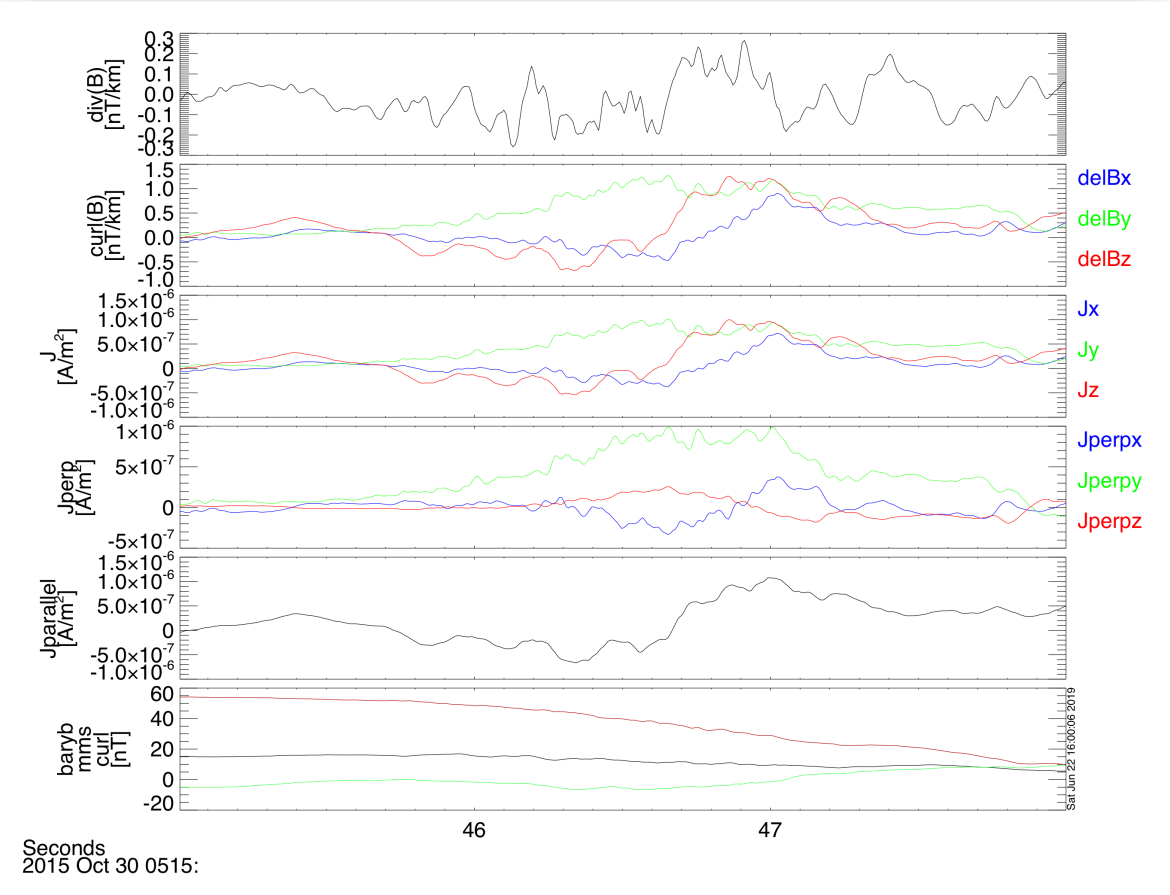




Analysis Tools

Curlometer technique

```
trange = ['2015-10-30/05:15:45', '2015-10-30/05:15:48']

mms_load_fgm, trange=trange, /get_fgm_ephemeris, probes=[1, 2, 3, 4], data_rate='brst'

fields = 'mms'+['1', '2', '3', '4']+'_fgm_b_gse_brst_l2'
positions = 'mms'+['1', '2', '3', '4']+'_fgm_r_gse_brst_l2'

mms_curl, trange=trange, fields=fields, positions=positions, suffix='_mms_curl'

tplot, ['divB','curlB','jtotal','jperp','jpar','baryb']+'_mms_curl'
```



Analysis Tools

More Examples

Basic:
projects/mms/examples/basic/

Advanced:
projects/mms/examples/advanced/

Getting Started

Requirements

Python 3.5+

Required packages:
pytplot
cdflib
pydivide
pyqtgraph
xarray
numpy
requests
dateutil

“pip install -r requirements.txt”
in the pyspedas folder should install everything needed.

BETA

What's New / Plug-in Status
IDL
    Getting Started
    Loading and Plotting Data
    Analysis Tools
Python
    Getting Started
    Loading and Plotting Data
    Current Status

Getting Started

Installing pySPEDAS

Bleeding edge: https://github.com/spedas/pyspedas

Config settings (e.g., local_data_dir) are set in the hash table stored in mms_config.py

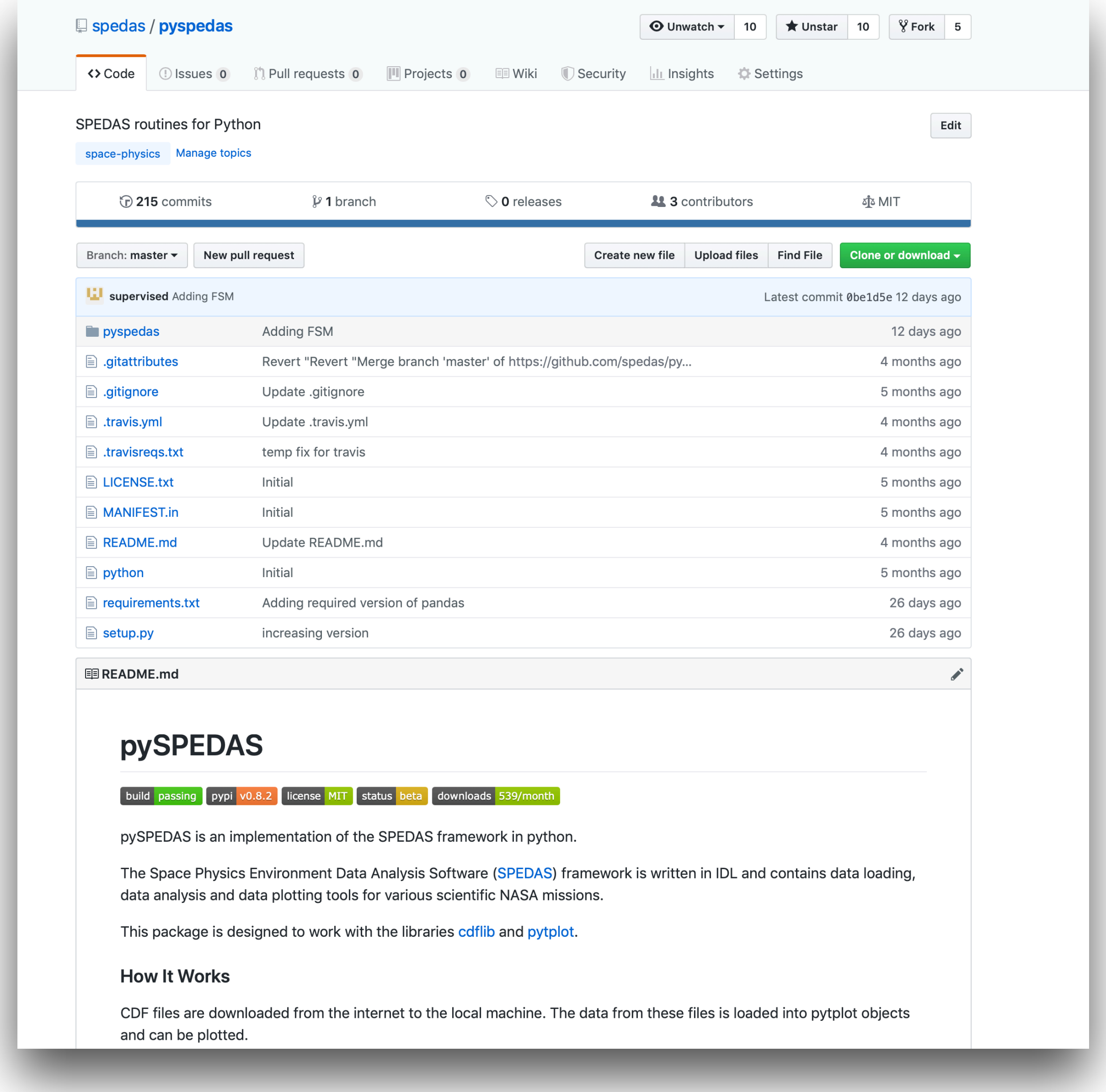
pip install pyspedas

or

pip install pyspedas --upgrade

to get the latest version

BETA



SPEDAS routines for Python

supervised Adding FSM — Latest commit 0be1d5e 12 days ago

pyspedas	Adding FSM	12 days ago
.gitattributes	Revert "Revert "Merge branch 'master' of https://github.com/spedas/py...	4 months ago
.gitignore	Update .gitignore	5 months ago
.travis.yml	Update .travis.yml	4 months ago
.travisreqs.txt	temp fix for travis	4 months ago
LICENSE.txt	Initial	5 months ago
MANIFEST.in	Initial	5 months ago
README.md	Update README.md	4 months ago
python	Initial	5 months ago
requirements.txt	Adding required version of pandas	26 days ago
setup.py	increasing version	26 days ago

README.md

pySPEDAS

build passing | pypi v0.8.2 | license MIT | status beta | downloads 539/month

pySPEDAS is an implementation of the SPEDAS framework in python.

The Space Physics Environment Data Analysis Software (SPEDAS) framework is written in IDL and contains data loading, data analysis and data plotting tools for various scientific NASA missions.

This package is designed to work with the libraries cdflib and pytplot.

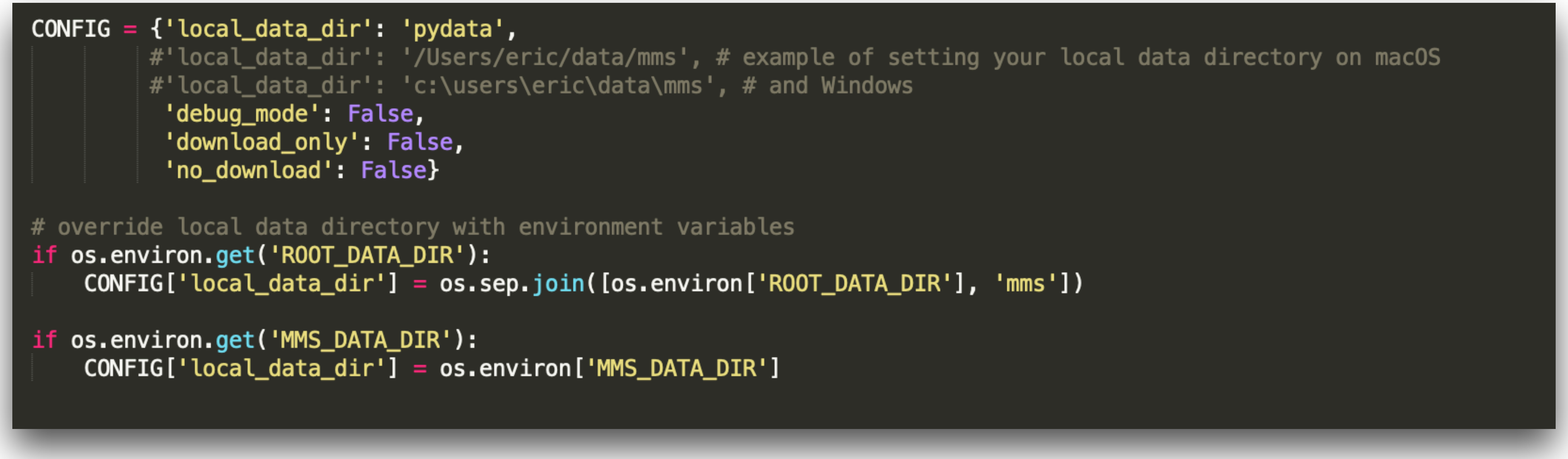
How It Works

CDF files are downloaded from the internet to the local machine. The data from these files is loaded into pytplot objects and can be plotted.

What's New / Plug-in Status
IDL
    Getting Started
    Loading and Plotting Data
    Analysis Tools
Python
    Getting Started
    Loading and Plotting Data
    Current Status

Getting Started

Configuring MMS in pySPEDAS



```
CONFIG = {'local_data_dir': 'pydata',
         #'local_data_dir': '/Users/eric/data/mms', # example of setting your local data directory on macOS
         #'local_data_dir': 'c:\users\eric\data\mms', # and Windows
          'debug_mode': False,
          'download_only': False,
          'no_download': False}

# override local data directory with environment variables
if os.environ.get('ROOT_DATA_DIR'):
    CONFIG['local_data_dir'] = os.sep.join([os.environ['ROOT_DATA_DIR'], 'mms'])

if os.environ.get('MMS_DATA_DIR'):
    CONFIG['local_data_dir'] = os.environ['MMS_DATA_DIR']
```

BETA

Getting Started

MMS Load Routines

mms_load_fgm	Fluxgate Magnetometer
mms_load_scm	Search-coil Magnetometer
mms_load_fsm	L3 FGM+SCM
mms_load_mec	Ephemeris and Coordinates
mms_load_fpi	Fast Plasma Investigation
mms_load_hpca	Hot Plasma Composition Analyzer
mms_load_eis	Energetic Ion Spectrometer
mms_load_feeps	Fly’s Eye Energetic Particle Sensor
mms_load_edp	Electric-field Double Probe
mms_load_edi	Electron Drift Instrument
mms_load_dsp	Digital Signal Processor
mms_load_aspoc	Active Spacecraft Potential Control

BETA

Getting Started

Standard Keywords

- trange
- probe
- level
- data_rate
- datatype

```
trange=['2015-10-16', '2015-10-17']

probes=[1, 2, 3, 4]

level='l2'

data_rate='srvy'

datatype=['des-moms', 'dis-moms']
```

Getting Started

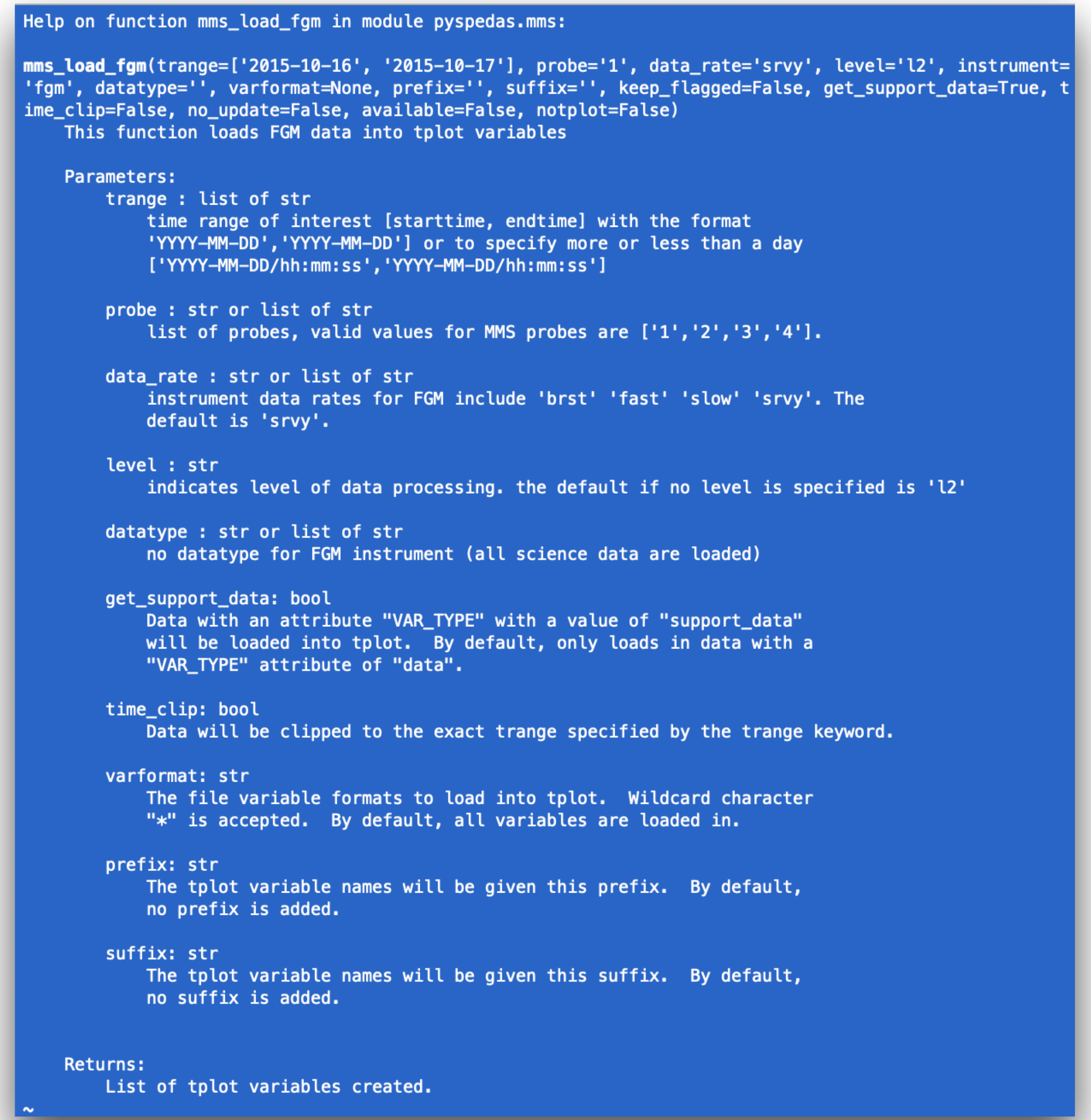
Standard Keywords

• suffix	`suffix='_burst_mode'`
• time_clip	`time_clip=True`
• no_update	`no_update=True`
• notplot	`notplot=True`
• varformat	`varformat=‘*_fgm_*’`

Getting Started

Standard Keywords

For a complete list of keywords and their descriptions, use the ‘help’ command, e.g., help(mms_load_fgm):

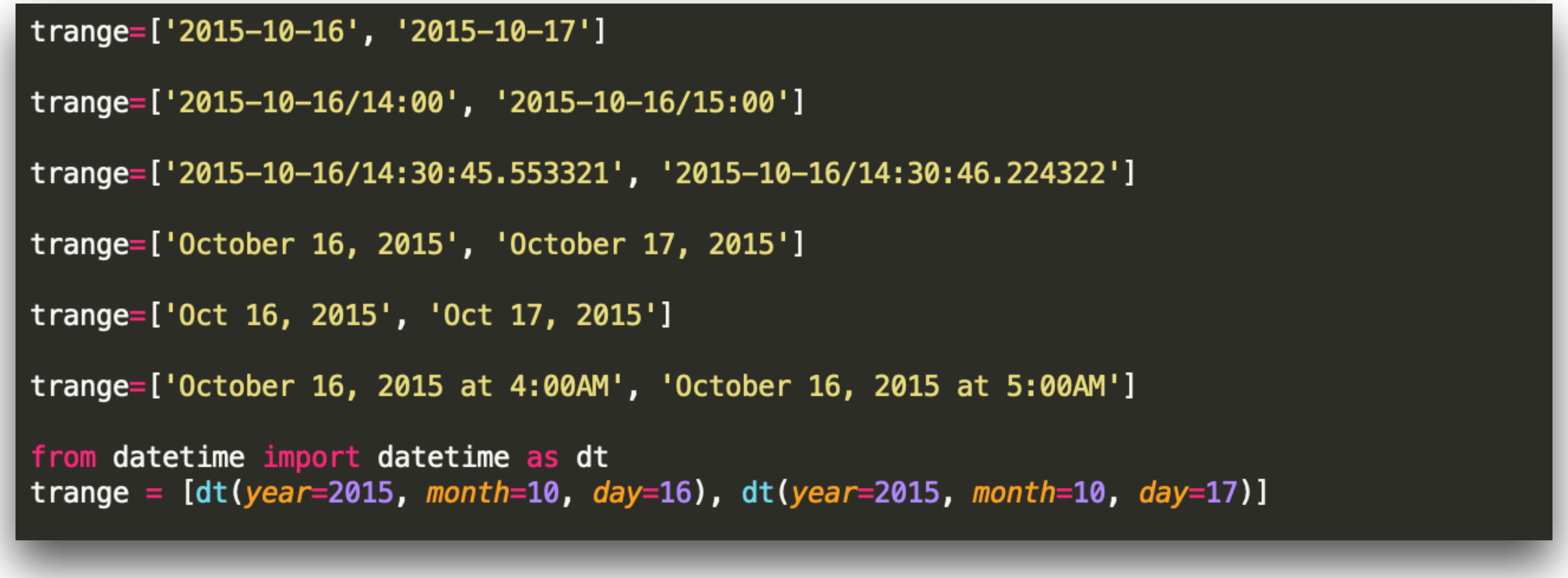


```
Help on function mms_load_fgm in module pyspedas.mms:

mms_load_fgm(trange=['2015-10-16', '2015-10-17'], probe='1', data_rate='srvy', level='l2', instrument=
'fgm', datatype='', varformat=None, prefix='', suffix='', keep_flagged=False, get_support_data=True, t
ime_clip=False, no_update=False, available=False, notplot=False)
    This function loads FGM data into tplot variables

    Parameters:
        trange : list of str
            time range of interest [starttime, endtime] with the format
            'YYYY-MM-DD','YYYY-MM-DD'] or to specify more or less than a day
            ['YYYY-MM-DD/hh:mm:ss','YYYY-MM-DD/hh:mm:ss']

        probe : str or list of str
            list of probes, valid values for MMS probes are ['1','2','3','4'].

        data_rate : str or list of str
            instrument data rates for FGM include 'brst' 'fast' 'slow' 'srvy'. The
            default is 'srvy'.

        level : str
            indicates level of data processing. the default if no level is specified is 'l2'

        datatype : str or list of str
            no datatype for FGM instrument (all science data are loaded)

        get_support_data: bool
            Data with an attribute "VAR_TYPE" with a value of "support_data"
            will be loaded into tplot.  By default, only loads in data with a
            "VAR_TYPE" attribute of "data".

        time_clip: bool
            Data will be clipped to the exact trange specified by the trange keyword.

        varformat: str
            The file variable formats to load into tplot.  Wildcard character
            "*" is accepted.  By default, all variables are loaded in.

        prefix: str
            The tplot variable names will be given this prefix.  By default,
            no prefix is added.

        suffix: str
            The tplot variable names will be given this suffix.  By default,
            no suffix is added.


    Returns:
        List of tplot variables created.
~
```

Getting Started

Note on trange

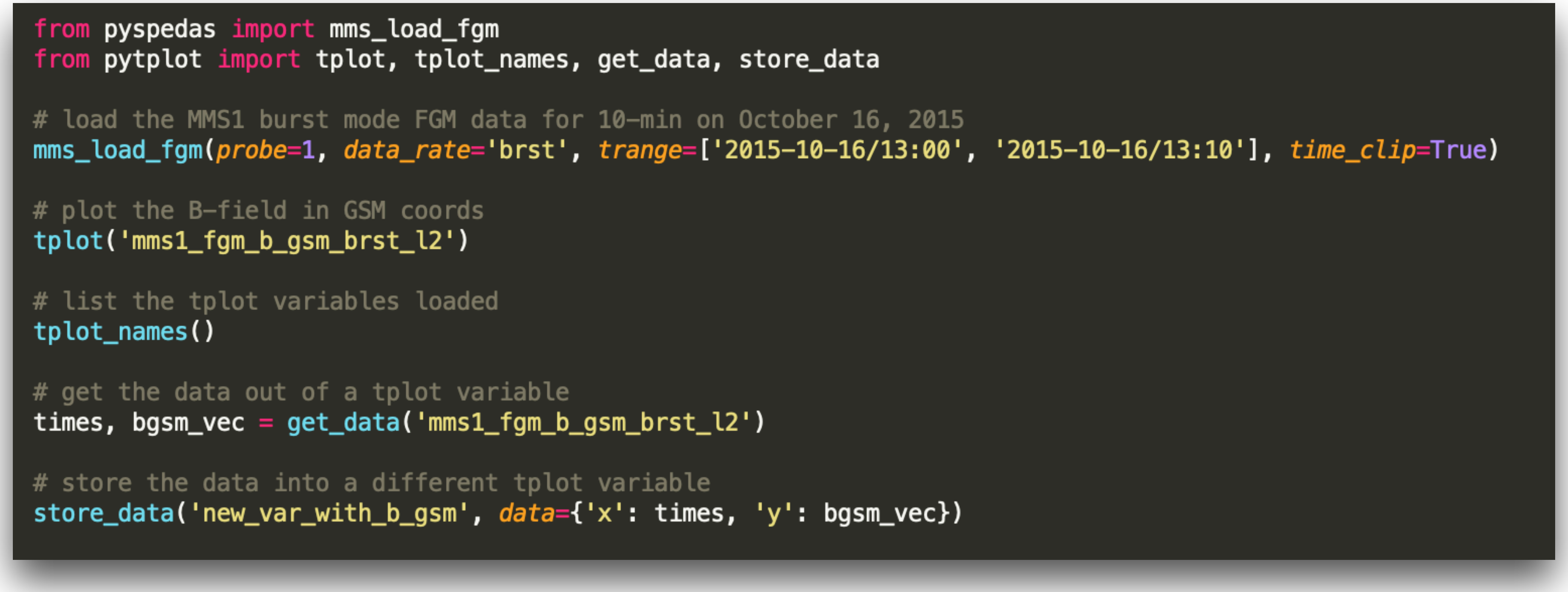
The **trange** keyword accepts a wide range of different formats:



```
trange=['2015-10-16', '2015-10-17']

trange=['2015-10-16/14:00', '2015-10-16/15:00']

trange=['2015-10-16/14:30:45.553321', '2015-10-16/14:30:46.224322']

trange=['October 16, 2015', 'October 17, 2015']

trange=['Oct 16, 2015', 'Oct 17, 2015']

trange=['October 16, 2015 at 4:00AM', 'October 16, 2015 at 5:00AM']

from datetime import datetime as dt
trange = [dt(year=2015, month=10, day=16), dt(year=2015, month=10, day=17)]
```

Loading and Plotting Data

Simple Example



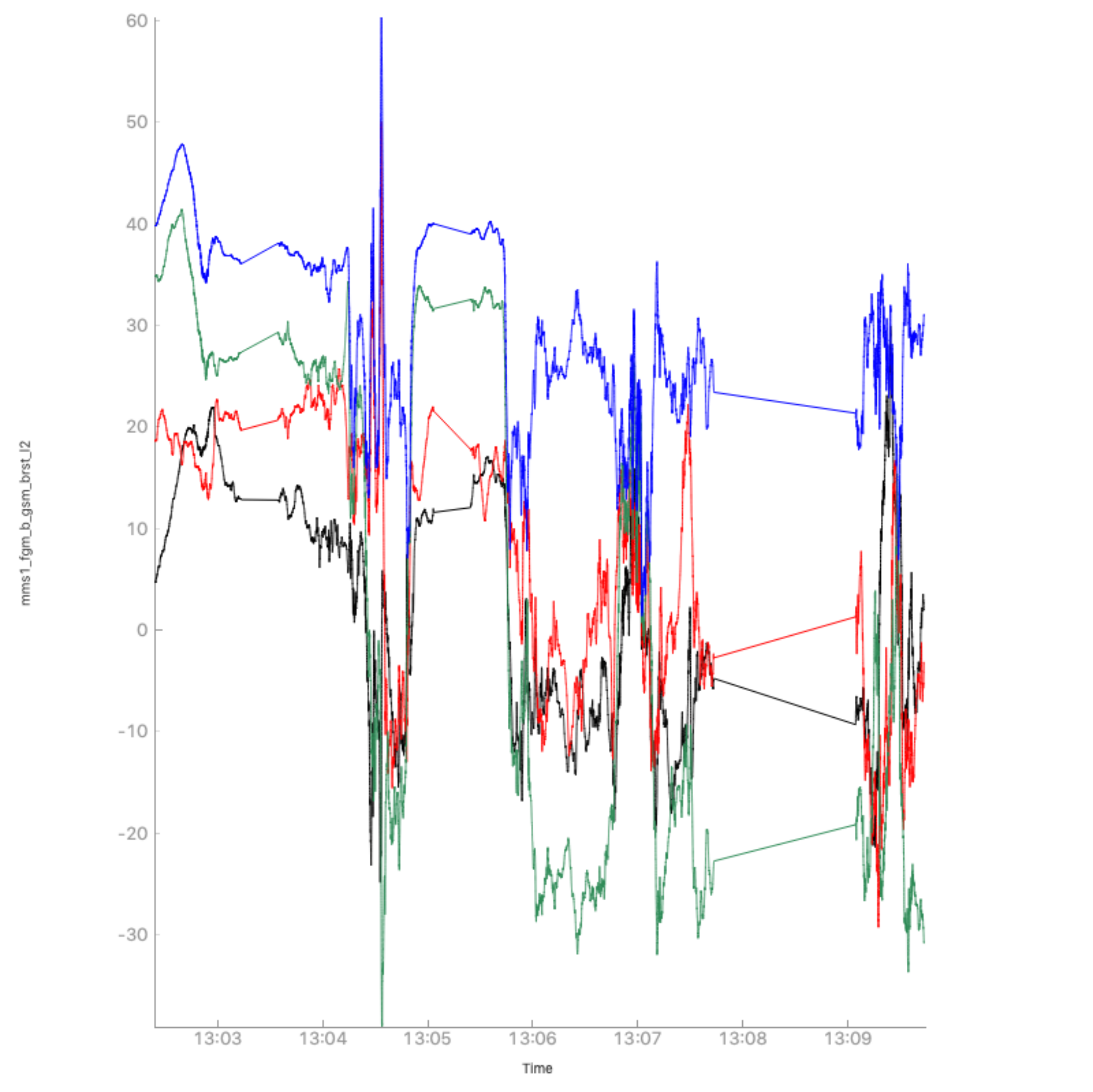
```
from pyspedas import mms_load_fgm
from pytplot import tplot, tplot_names, get_data, store_data

# load the MMS1 burst mode FGM data for 10-min on October 16, 2015
mms_load_fgm(probe=1, data_rate='brst', trange=['2015-10-16/13:00', '2015-10-16/13:10'], time_clip=True)

# plot the B-field in GSM coords
tplot('mms1_fgm_b_gsm_brst_l2')

# list the tplot variables loaded
tplot_names()

# get the data out of a tplot variable
times, bgsm_vec = get_data('mms1_fgm_b_gsm_brst_l2')

# store the data into a different tplot variable
store_data('new_var_with_b_gsm', data={'x': times, 'y': bgsm_vec})
```

What's New / Plug-in Status
IDL
- Getting Started
- Loading and Plotting Data
- Analysis Tools

Python
- Getting Started
- **Loading and Plotting Data**
- Current Status

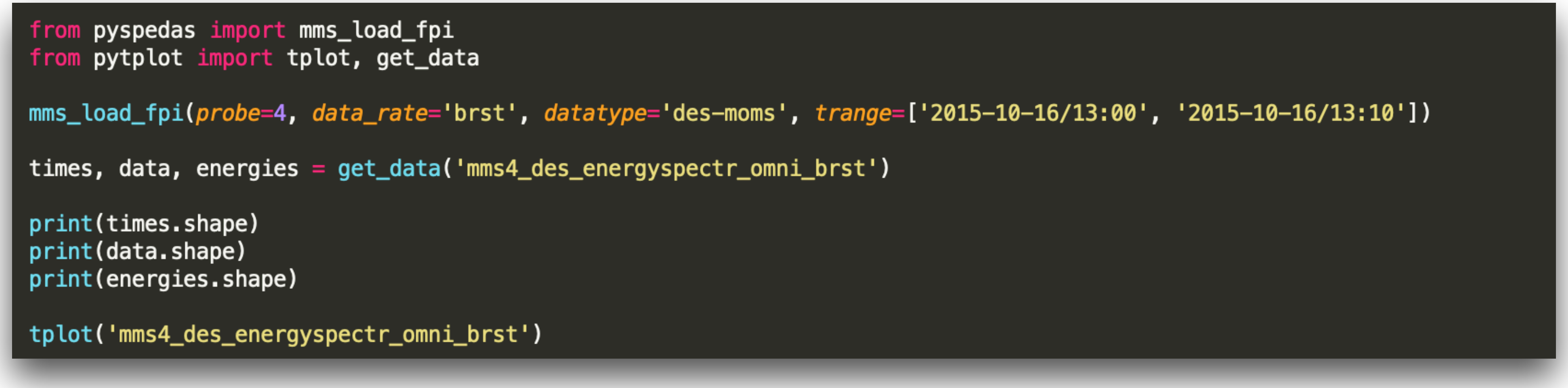
Loading and Plotting Data

Simple Example

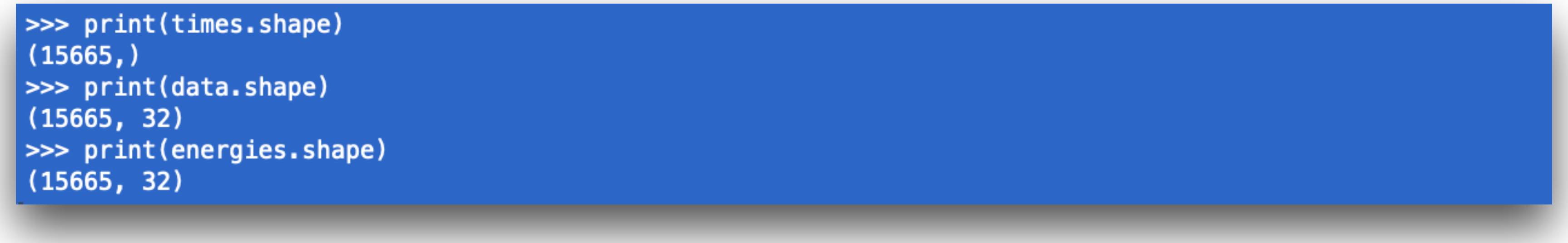


Loading and Plotting Data

FPI



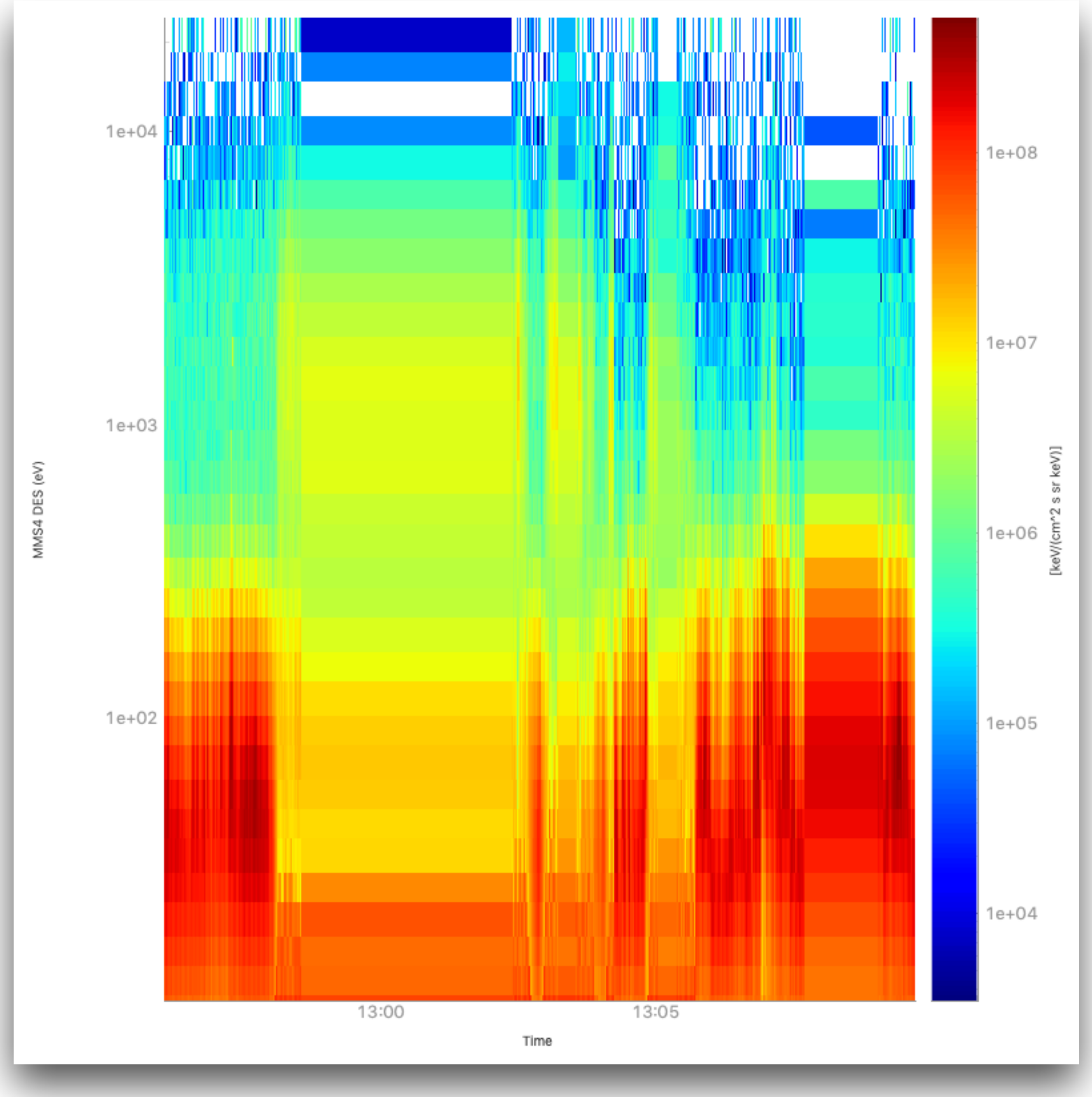
```python
from pyspedas import mms_load_fpi
from pytplot import tplot, get_data

mms_load_fpi(probe=4, data_rate='brst', datatype='des-moms', trange=['2015-10-16/13:00', '2015-10-16/13:10'])

times, data, energies = get_data('mms4_des_energyspectr_omni_brst')

print(times.shape)
print(data.shape)
print(energies.shape)

tplot('mms4_des_energyspectr_omni_brst')
```



```
>>> print(times.shape)
(15665,)
>>> print(data.shape)
(15665, 32)
>>> print(energies.shape)
(15665, 32)
```

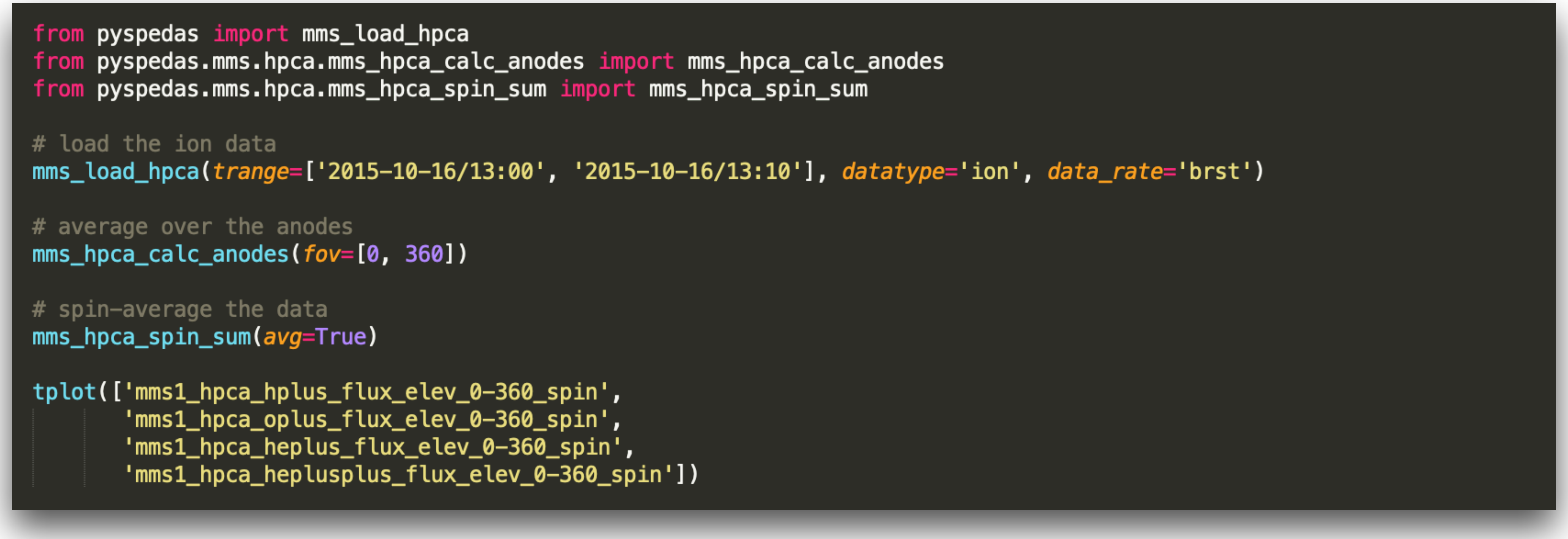
Loading and Plotting Data

FPI



Loading and Plotting Data

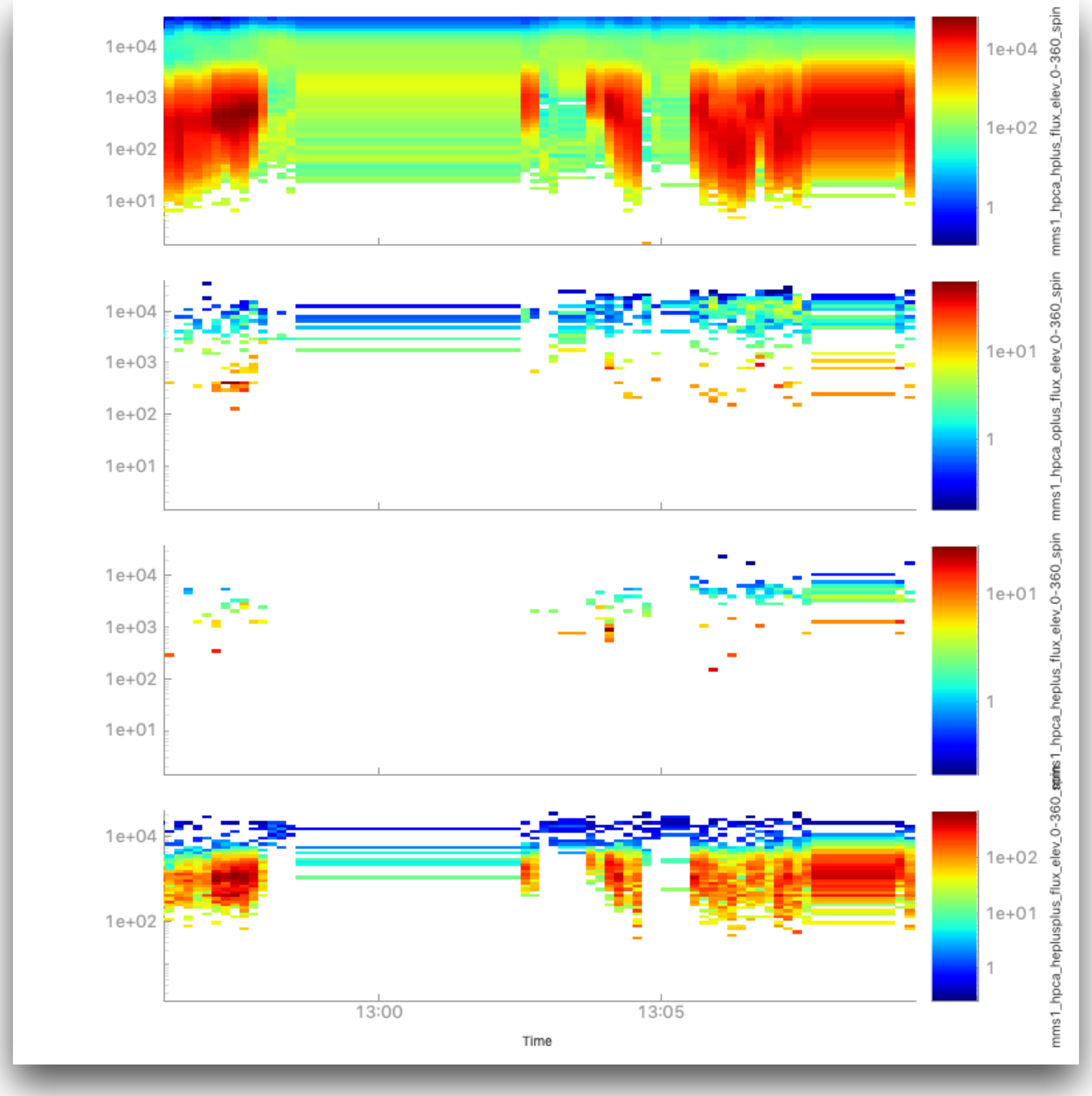
HPCA



```
from pyspedas import mms_load_hpca
from pyspedas.mms.hpca.mms_hpca_calc_anodes import mms_hpca_calc_anodes
from pyspedas.mms.hpca.mms_hpca_spin_sum import mms_hpca_spin_sum

# load the ion data
mms_load_hpca(trange=['2015-10-16/13:00', '2015-10-16/13:10'], datatype='ion', data_rate='brst')

# average over the anodes
mms_hpca_calc_anodes(fov=[0, 360])

# spin-average the data
mms_hpca_spin_sum(avg=True)

tplot(['mms1_hpca_hplus_flux_elev_0-360_spin',
       'mms1_hpca_oplus_flux_elev_0-360_spin',
       'mms1_hpca_heplus_flux_elev_0-360_spin',
       'mms1_hpca_heplusplus_flux_elev_0-360_spin'])
```

Loading and Plotting Data

HPCA

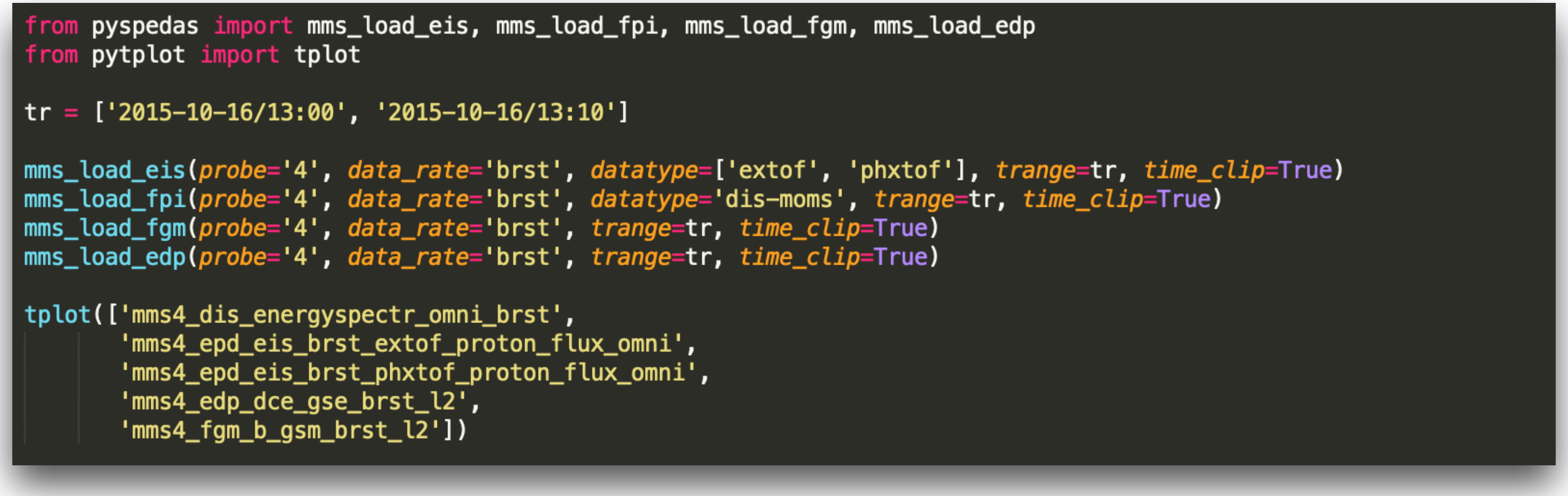


What's New / Plug-in Status
IDL
- Getting Started
- Loading and Plotting Data
- Analysis Tools

Python
- Getting Started
- **Loading and Plotting Data**
- Current Status

Loading and Plotting Data

Multiple Instruments



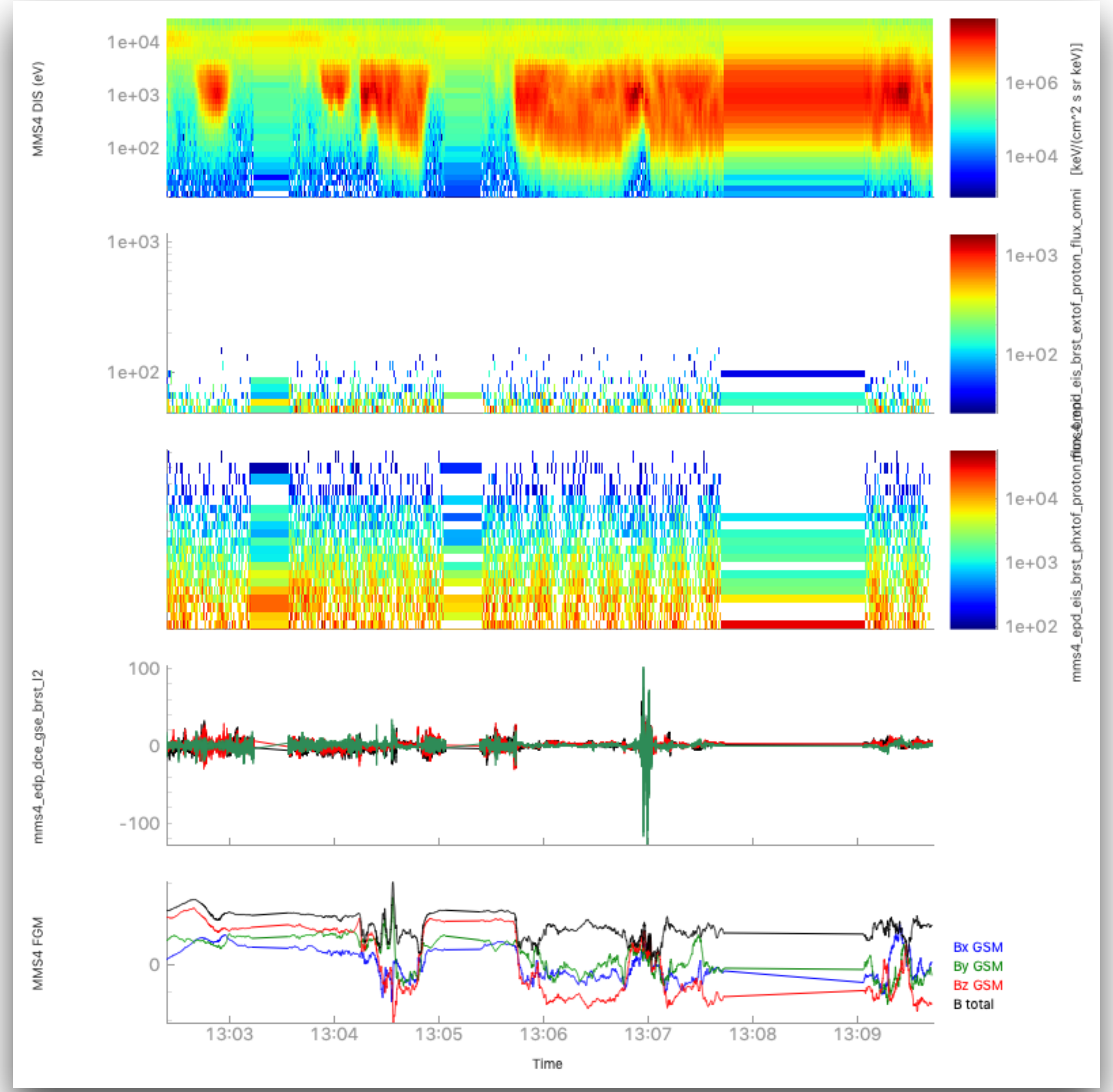
```
from pyspedas import mms_load_eis, mms_load_fpi, mms_load_fgm, mms_load_edp
from pytplot import tplot

tr = ['2015-10-16/13:00', '2015-10-16/13:10']

mms_load_eis(probe='4', data_rate='brst', datatype=['extof', 'phxtof'], trange=tr, time_clip=True)
mms_load_fpi(probe='4', data_rate='brst', datatype='dis-moms', trange=tr, time_clip=True)
mms_load_fgm(probe='4', data_rate='brst', trange=tr, time_clip=True)
mms_load_edp(probe='4', data_rate='brst', trange=tr, time_clip=True)

tplot(['mms4_dis_energyspectr_omni_brst',
       'mms4_epd_eis_brst_extof_proton_flux_omni',
       'mms4_epd_eis_brst_phxtof_proton_flux_omni',
       'mms4_edp_dce_gse_brst_l2',
       'mms4_fgm_b_gsm_brst_l2'])
```

What's New / Plug-in Status
IDL
- Getting Started
- Loading and Plotting Data
- Analysis Tools

Python
- Getting Started
- **Loading and Plotting Data**
- Current Status

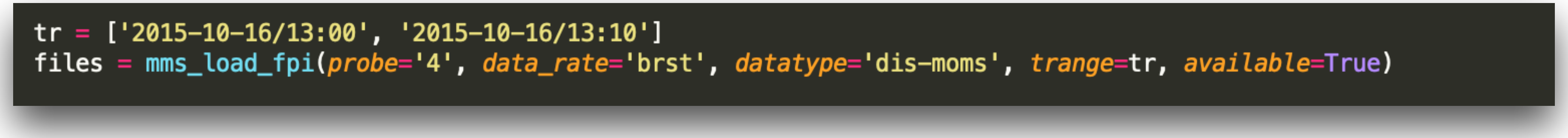
Loading and Plotting Data

Multiple Instruments

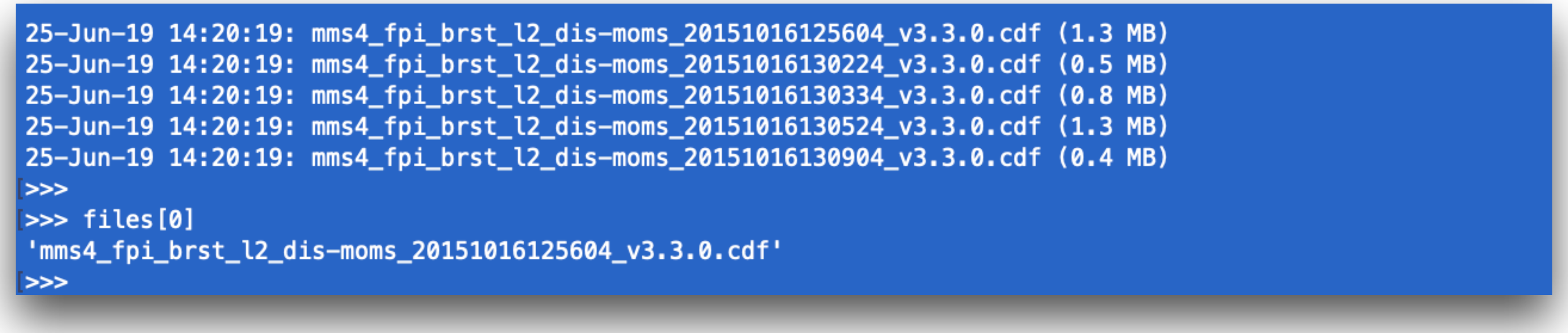


Loading and Plotting Data

Find available files



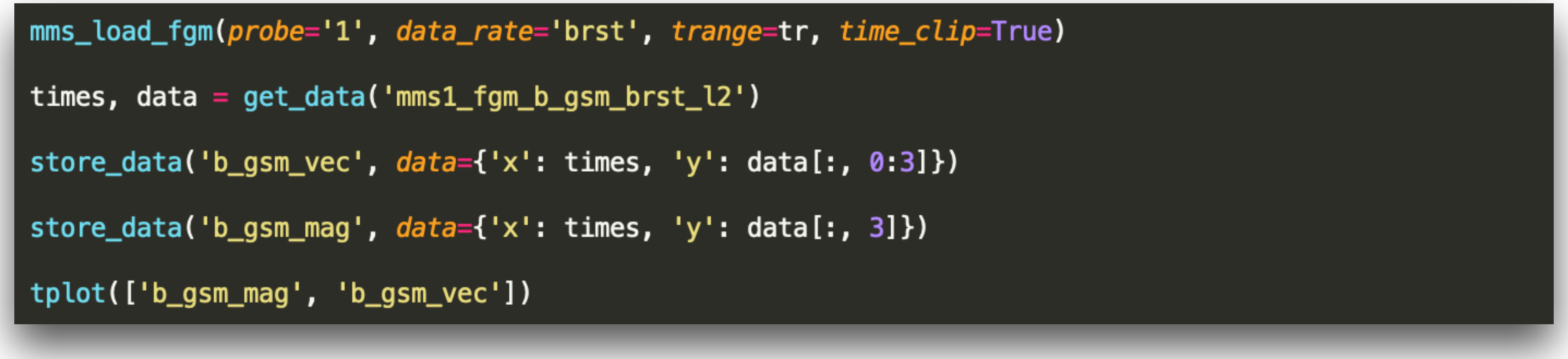
```
tr = ['2015-10-16/13:00', '2015-10-16/13:10']
files = mms_load_fpi(probe='4', data_rate='brst', datatype='dis-moms', trange=tr, available=True)
```



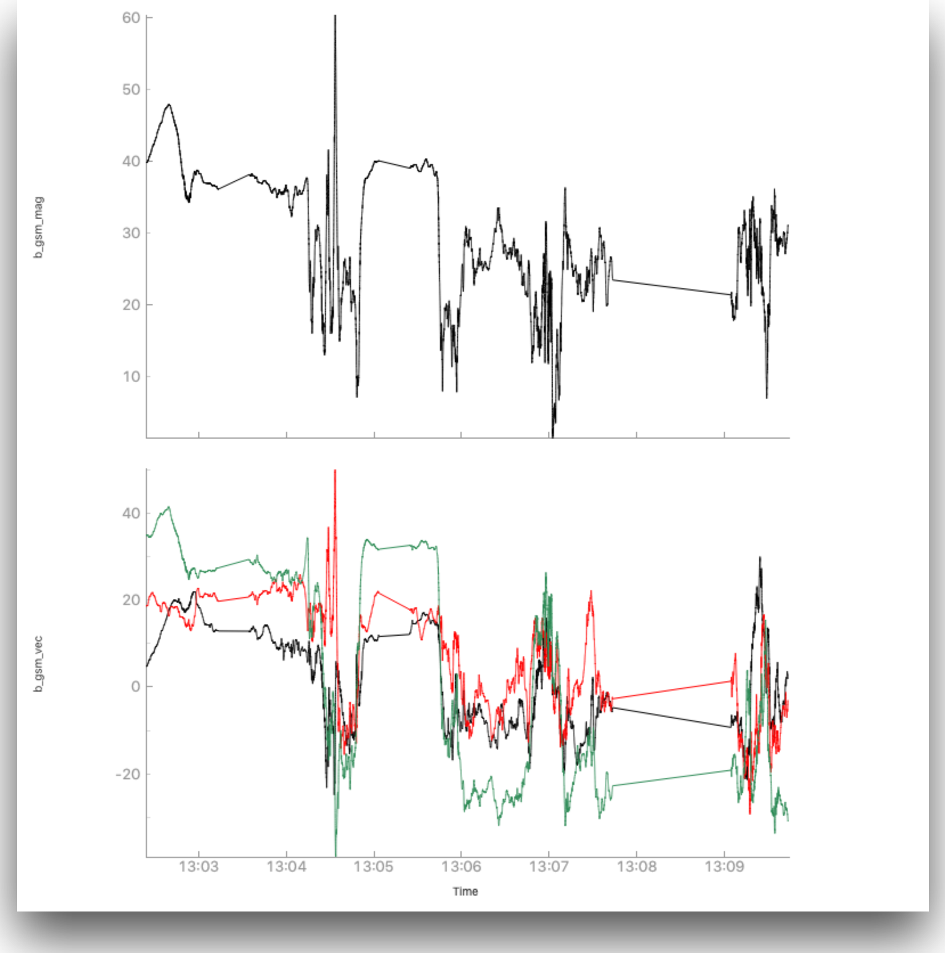
```
25-Jun-19 14:20:19: mms4_fpi_brst_l2_dis-moms_20151016125604_v3.3.0.cdf (1.3 MB)
25-Jun-19 14:20:19: mms4_fpi_brst_l2_dis-moms_20151016130224_v3.3.0.cdf (0.5 MB)
25-Jun-19 14:20:19: mms4_fpi_brst_l2_dis-moms_20151016130334_v3.3.0.cdf (0.8 MB)
25-Jun-19 14:20:19: mms4_fpi_brst_l2_dis-moms_20151016130524_v3.3.0.cdf (1.3 MB)
25-Jun-19 14:20:19: mms4_fpi_brst_l2_dis-moms_20151016130904_v3.3.0.cdf (0.4 MB)
>>>
>>> files[0]
'mms4_fpi_brst_l2_dis-moms_20151016125604_v3.3.0.cdf'
>>>
```

Loading and Plotting Data

Create pyTplot Variables

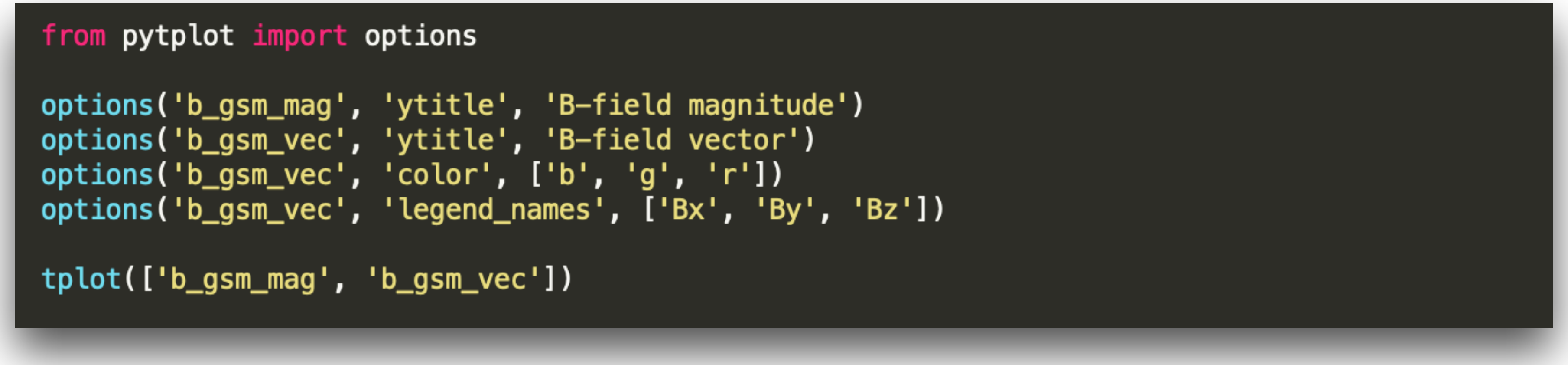


```
mms_load_fgm(probe='1', data_rate='brst', trange=tr, time_clip=True)

times, data = get_data('mms1_fgm_b_gsm_brst_l2')

store_data('b_gsm_vec', data={'x': times, 'y': data[:, 0:3]})

store_data('b_gsm_mag', data={'x': times, 'y': data[:, 3]})

tplot(['b_gsm_mag', 'b_gsm_vec'])
```

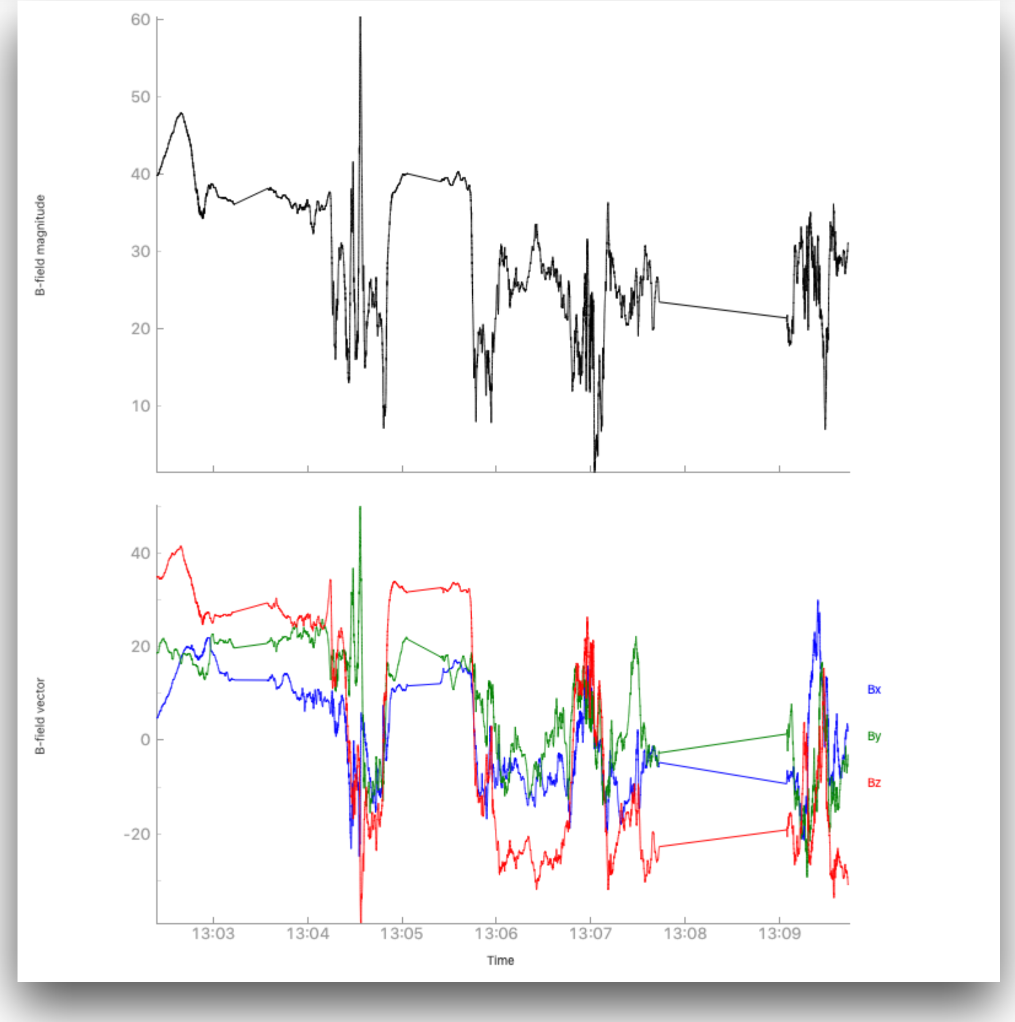


Loading and Plotting Data

Modify Variable Metadata

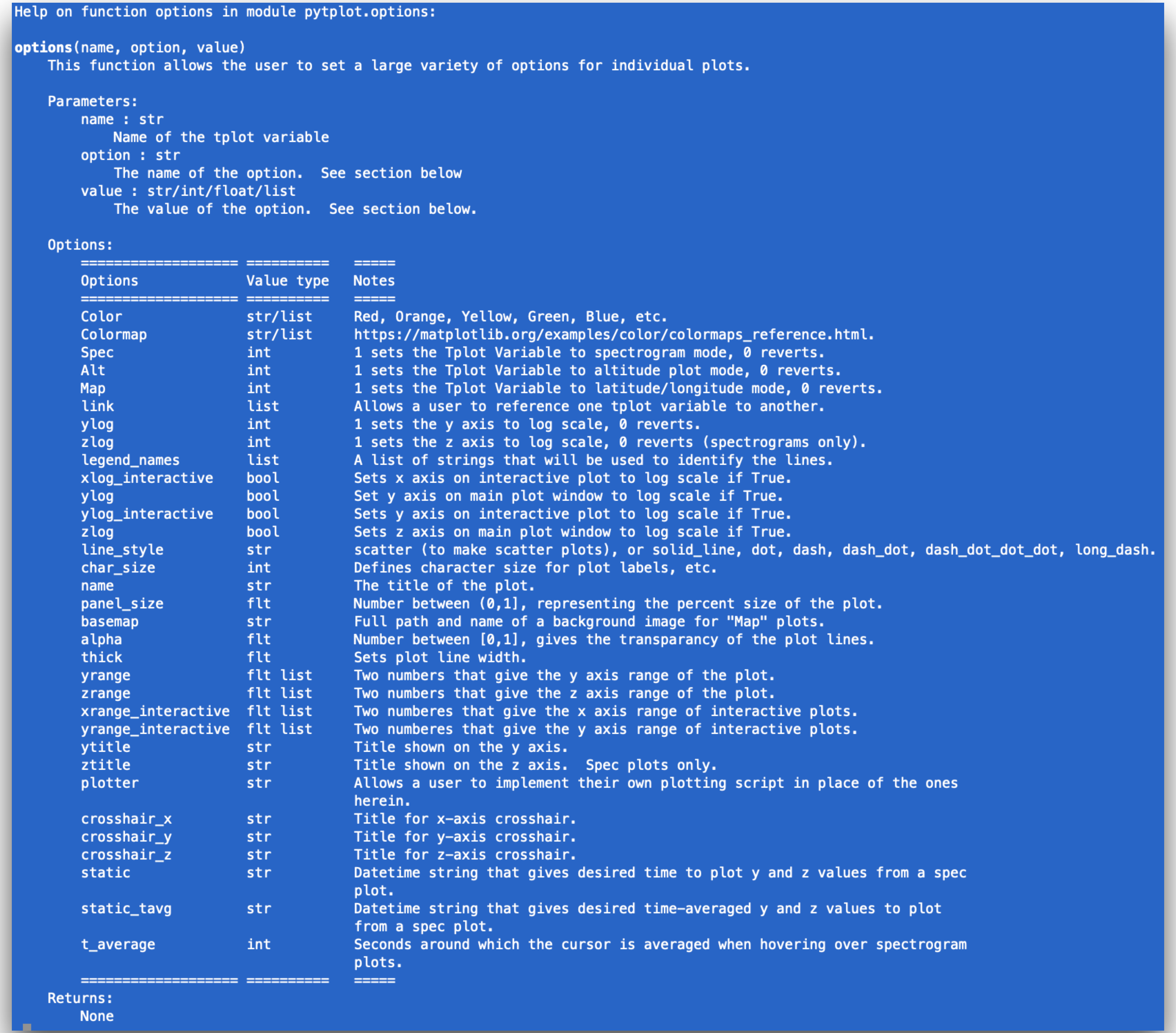


```
from pytplot import options

options('b_gsm_mag', 'ytitle', 'B-field magnitude')
options('b_gsm_vec', 'ytitle', 'B-field vector')
options('b_gsm_vec', 'color', ['b', 'g', 'r'])
options('b_gsm_vec', 'legend_names', ['Bx', 'By', 'Bz'])

tplot(['b_gsm_mag', 'b_gsm_vec'])
```



Loading and Plotting Data

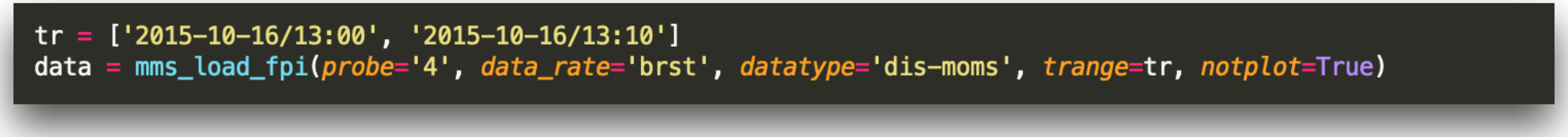
Modify Variable Metadata



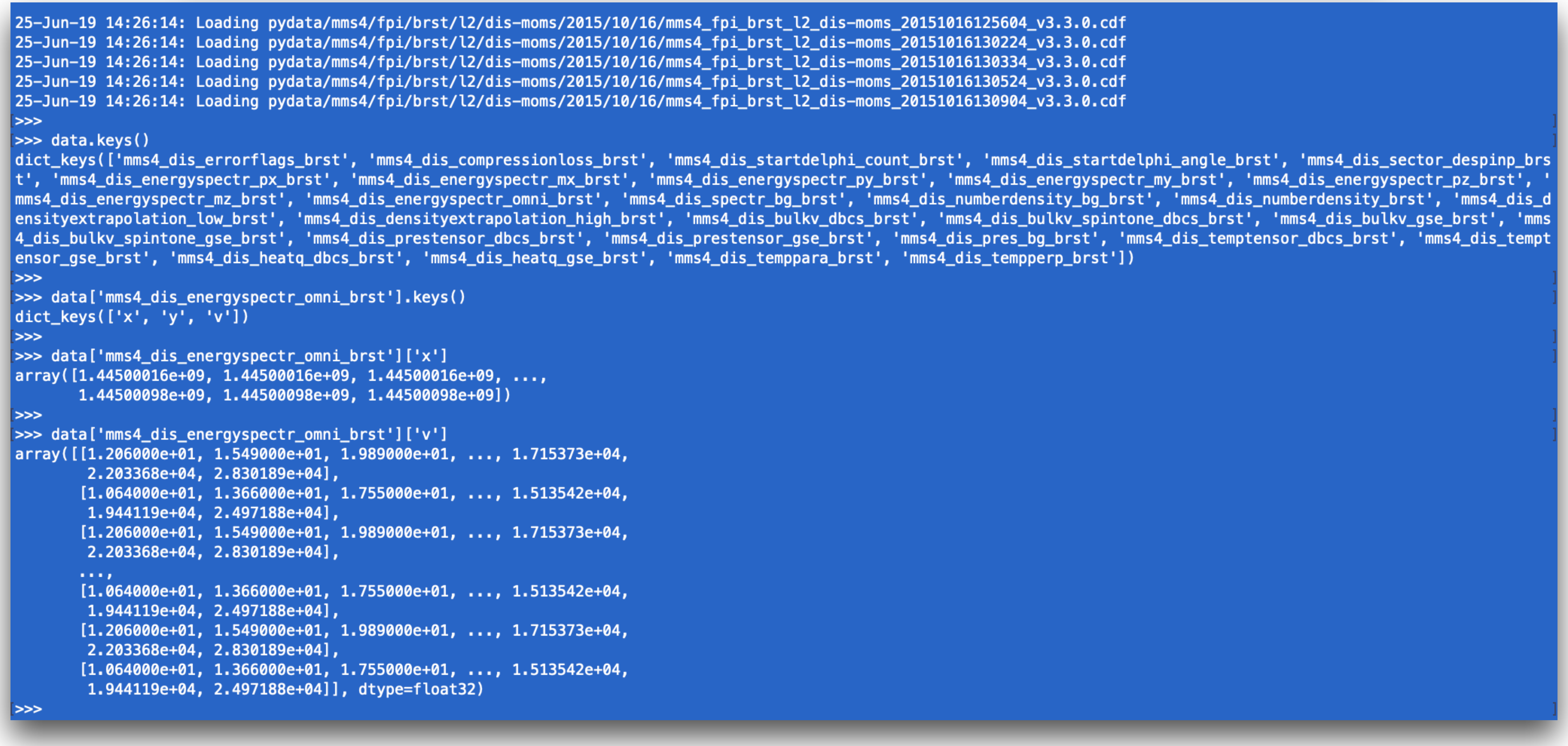
```
Help on function options in module pytplot.options:

options(name, option, value)
    This function allows the user to set a large variety of options for individual plots.

    Parameters:
        name : str
            Name of the tplot variable
        option : str
            The name of the option.  See section below
        value : str/int/float/list
            The value of the option.  See section below.

    Options:
        =================== ==========   =====
        Options             Value type   Notes
        =================== ==========   =====
        Color               str/list     Red, Orange, Yellow, Green, Blue, etc.
        Colormap            str/list     https://matplotlib.org/examples/color/colormaps_reference.html.
        Spec                int          1 sets the Tplot Variable to spectrogram mode, 0 reverts.
        Alt                 int          1 sets the Tplot Variable to altitude plot mode, 0 reverts.
        Map                 int          1 sets the Tplot Variable to latitude/longitude mode, 0 reverts.
        link                list         Allows a user to reference one tplot variable to another.
        ylog                int          1 sets the y axis to log scale, 0 reverts.
        zlog                int          1 sets the z axis to log scale, 0 reverts (spectrograms only).
        legend_names        list         A list of strings that will be used to identify the lines.
        xlog_interactive    bool         Sets x axis on interactive plot to log scale if True.
        ylog                bool         Set y axis on main plot window to log scale if True.
        ylog_interactive    bool         Sets y axis on interactive plot to log scale if True.
        zlog                bool         Sets z axis on main plot window to log scale if True.
        line_style          str          scatter (to make scatter plots), or solid_line, dot, dash, dash_dot, dash_dot_dot_dot, long_dash.
        char_size           int          Defines character size for plot labels, etc.
        name                str          The title of the plot.
        panel_size          flt          Number between (0,1], representing the percent size of the plot.
        basemap             str          Full path and name of a background image for "Map" plots.
        alpha               flt          Number between [0,1], gives the transparancy of the plot lines.
        thick               flt          Sets plot line width.
        yrange              flt list     Two numbers that give the y axis range of the plot.
        zrange              flt list     Two numbers that give the z axis range of the plot.
        xrange_interactive  flt list     Two numberes that give the x axis range of interactive plots.
        yrange_interactive  flt list     Two numberes that give the y axis range of interactive plots.
        ytitle              str          Title shown on the y axis.
        ztitle              str          Title shown on the z axis.  Spec plots only.
        plotter             str          Allows a user to implement their own plotting script in place of the ones
                                         herein.
        crosshair_x         str          Title for x-axis crosshair.
        crosshair_y         str          Title for y-axis crosshair.
        crosshair_z         str          Title for z-axis crosshair.
        static              str          Datetime string that gives desired time to plot y and z values from a spec
                                         plot.
        static_tavg         str          Datetime string that gives desired time-averaged y and z values to plot
                                         from a spec plot.
        t_average           int          Seconds around which the cursor is averaged when hovering over spectrogram
                                         plots.
        =================== ==========   =====
    Returns:
        None
```

Loading and Plotting Data

Return the data without creating pyTplot variables



```
tr = ['2015-10-16/13:00', '2015-10-16/13:10']
data = mms_load_fpi(probe='4', data_rate='brst', datatype='dis-moms', trange=tr, notplot=True)
```

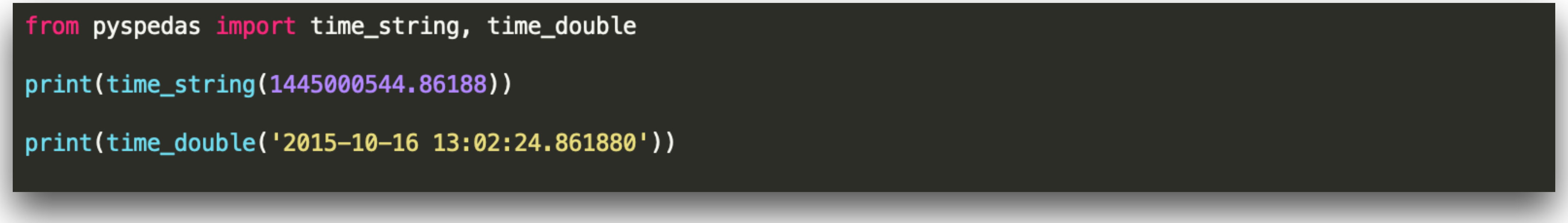


```
25-Jun-19 14:26:14: Loading pydata/mms4/fpi/brst/l2/dis-moms/2015/10/16/mms4_fpi_brst_l2_dis-moms_20151016125604_v3.3.0.cdf
25-Jun-19 14:26:14: Loading pydata/mms4/fpi/brst/l2/dis-moms/2015/10/16/mms4_fpi_brst_l2_dis-moms_20151016130224_v3.3.0.cdf
25-Jun-19 14:26:14: Loading pydata/mms4/fpi/brst/l2/dis-moms/2015/10/16/mms4_fpi_brst_l2_dis-moms_20151016130334_v3.3.0.cdf
25-Jun-19 14:26:14: Loading pydata/mms4/fpi/brst/l2/dis-moms/2015/10/16/mms4_fpi_brst_l2_dis-moms_20151016130524_v3.3.0.cdf
25-Jun-19 14:26:14: Loading pydata/mms4/fpi/brst/l2/dis-moms/2015/10/16/mms4_fpi_brst_l2_dis-moms_20151016130904_v3.3.0.cdf
>>>
>>> data.keys()
dict_keys(['mms4_dis_errorflags_brst', 'mms4_dis_compressionloss_brst', 'mms4_dis_startdelphi_count_brst', 'mms4_dis_startdelphi_angle_brst', 'mms4_dis_sector_despinp_brs
t', 'mms4_dis_energyspectr_px_brst', 'mms4_dis_energyspectr_mx_brst', 'mms4_dis_energyspectr_py_brst', 'mms4_dis_energyspectr_my_brst', 'mms4_dis_energyspectr_pz_brst', '
mms4_dis_energyspectr_mz_brst', 'mms4_dis_energyspectr_omni_brst', 'mms4_dis_spectr_bg_brst', 'mms4_dis_numberdensity_bg_brst', 'mms4_dis_numberdensity_brst', 'mms4_dis_d
ensityextrapolation_low_brst', 'mms4_dis_densityextrapolation_high_brst', 'mms4_dis_bulkv_dbcs_brst', 'mms4_dis_bulkv_spintone_dbcs_brst', 'mms4_dis_bulkv_gse_brst', 'mms
4_dis_bulkv_spintone_gse_brst', 'mms4_dis_prestensor_dbcs_brst', 'mms4_dis_prestensor_gse_brst', 'mms4_dis_pres_bg_brst', 'mms4_dis_temptensor_dbcs_brst', 'mms4_dis_tempt
ensor_gse_brst', 'mms4_dis_heatq_dbcs_brst', 'mms4_dis_heatq_gse_brst', 'mms4_dis_temppara_brst', 'mms4_dis_tempperp_brst'])
>>>
>>> data['mms4_dis_energyspectr_omni_brst'].keys()
dict_keys(['x', 'y', 'v'])
>>>
>>> data['mms4_dis_energyspectr_omni_brst']['x']
array([1.44500016e+09, 1.44500016e+09, 1.44500016e+09, ...,
       1.44500098e+09, 1.44500098e+09, 1.44500098e+09])
>>>
>>> data['mms4_dis_energyspectr_omni_brst']['v']
array([[1.206000e+01, 1.549000e+01, 1.989000e+01, ..., 1.715373e+04,
        2.203368e+04, 2.830189e+04],
       [1.064000e+01, 1.366000e+01, 1.755000e+01, ..., 1.513542e+04,
        1.944119e+04, 2.497188e+04],
       [1.206000e+01, 1.549000e+01, 1.989000e+01, ..., 1.715373e+04,
        2.203368e+04, 2.830189e+04],
       ...,
       [1.064000e+01, 1.366000e+01, 1.755000e+01, ..., 1.513542e+04,
        1.944119e+04, 2.497188e+04],
       [1.206000e+01, 1.549000e+01, 1.989000e+01, ..., 1.715373e+04,
        2.203368e+04, 2.830189e+04],
       [1.064000e+01, 1.366000e+01, 1.755000e+01, ..., 1.513542e+04,
        1.944119e+04, 2.497188e+04]], dtype=float32)
>>>
```

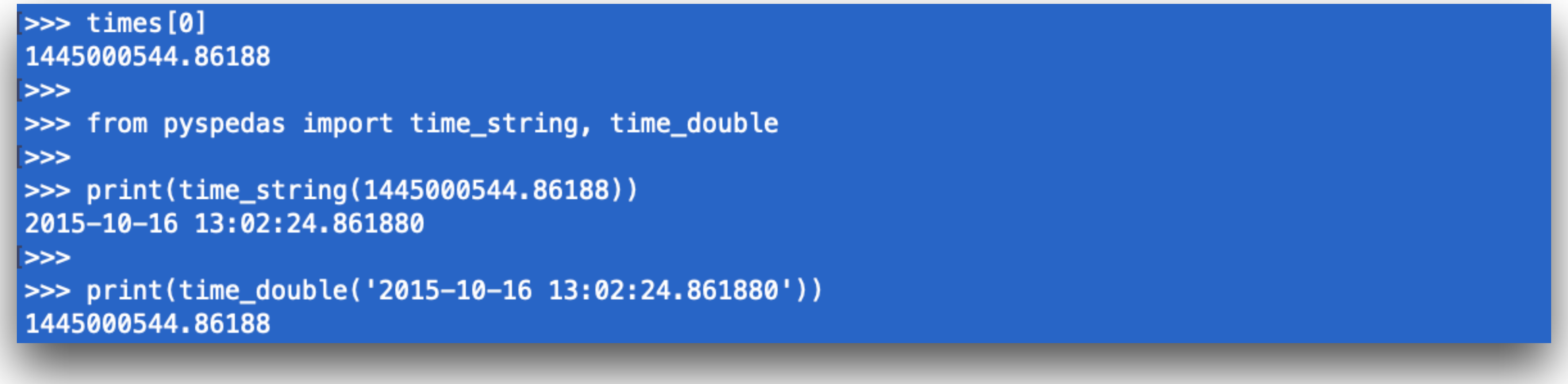
What's New / Plug-in Status
IDL
    Getting Started
    Loading and Plotting Data
    Analysis Tools
Python
    Getting Started
    Loading and Plotting Data
    Current Status

Loading and Plotting Data

Time Conversions



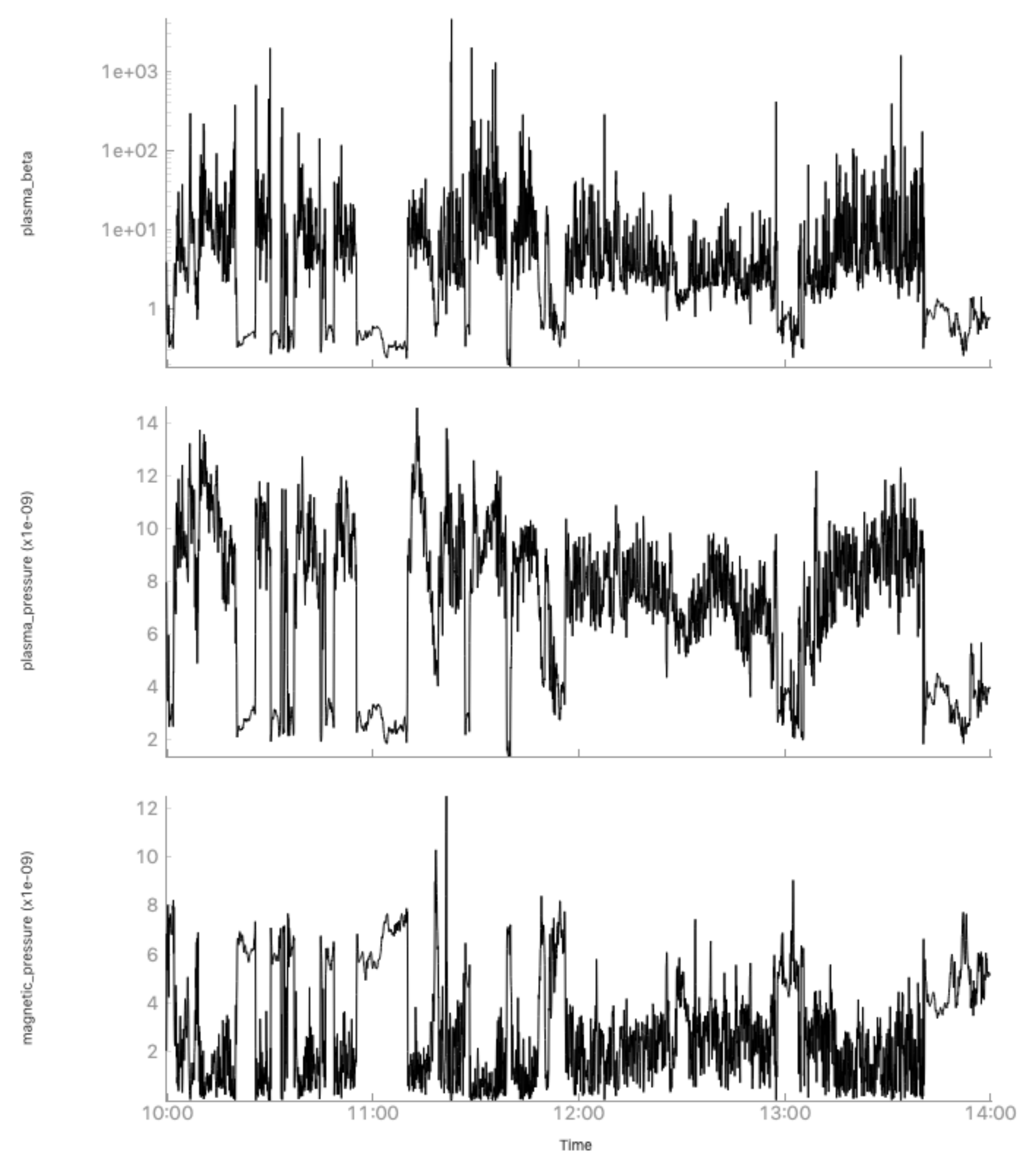
```
from pyspedas import time_string, time_double

print(time_string(1445000544.86188))

print(time_double('2015-10-16 13:02:24.861880'))
```



```
>>> times[0]
1445000544.86188
>>>
>>> from pyspedas import time_string, time_double
>>>
>>> print(time_string(1445000544.86188))
2015-10-16 13:02:24.861880
>>>
>>> print(time_double('2015-10-16 13:02:24.861880'))
1445000544.86188
```

Loading and Plotting Data

Plasma Beta with FPI and FGM



mms/examples/basic/mms_plasma_beta_crib.py

Loading and Plotting Data

PYTHONSTARTUP

To avoid manually importing the load routines, you can add the following file to your PYTHONSTARTUP environment variable:

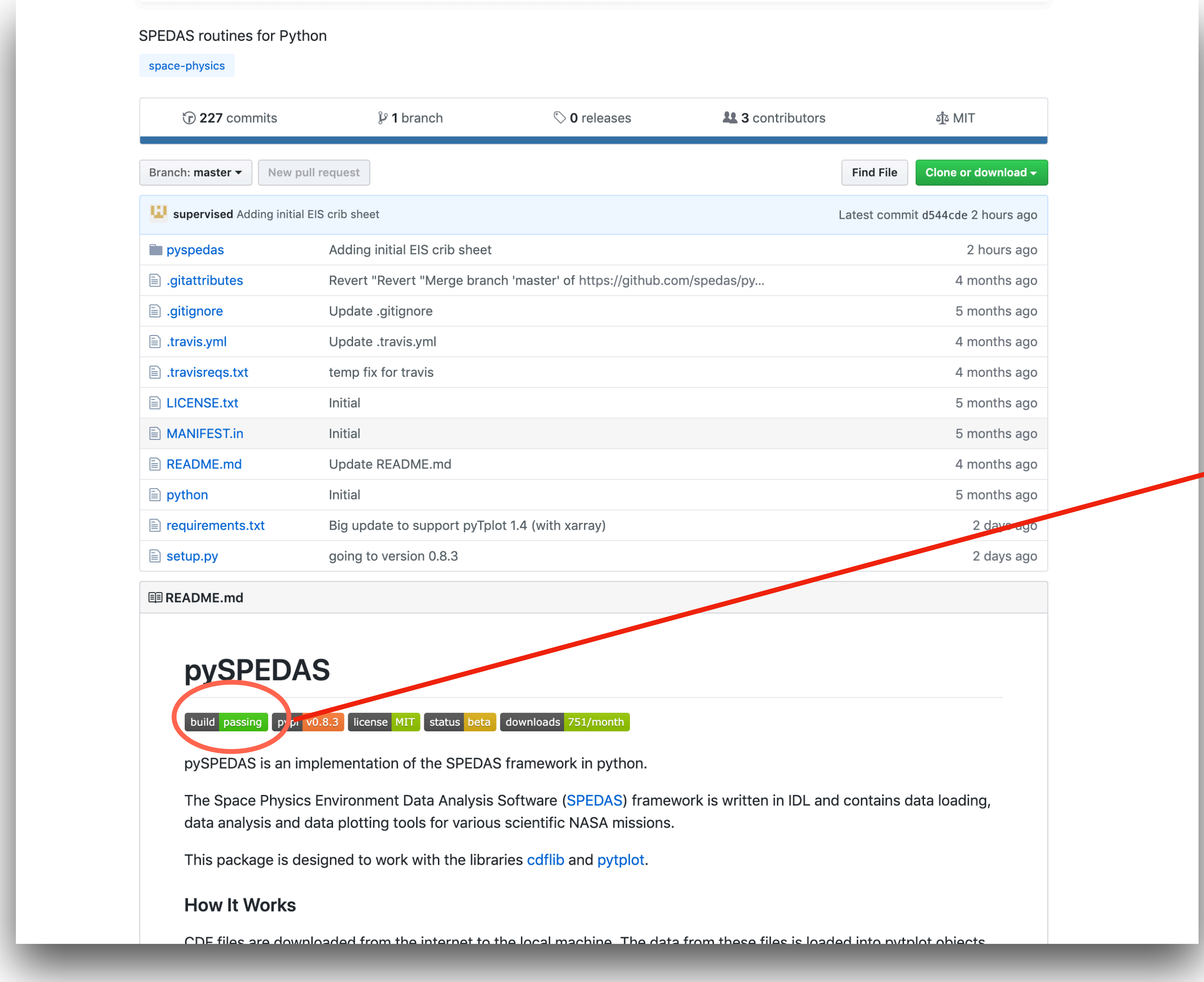
pyspedas/mms/mms_python_startup.py

What's New / Plug-in Status
IDL
Getting Started
Loading and Plotting Data
Analysis Tools
Python
Getting Started
Loading and Plotting Data
Current Status

Current Status

QA

- 20+ tests, automatically ran on every commit
- Tests mostly cover loading MMS data



SPEDAS routines for Python

space-physics

227 commits | 1 branch | 0 releases | 3 contributors | MIT

supervised Adding initial EIS crib sheet — Latest commit d544cde 2 hours ago

pyspedas	Adding initial EIS crib sheet	2 hours ago
.gitattributes	Revert "Revert "Merge branch 'master' of https://github.com/spedas/py…	4 months ago
.gitignore	Update .gitignore	5 months ago
.travis.yml	Update .travis.yml	4 months ago
.travisreqs.txt	temp fix for travis	4 months ago
LICENSE.txt	Initial	5 months ago
MANIFEST.in	Initial	5 months ago
README.md	Update README.md	4 months ago
python	Initial	5 months ago
requirements.txt	Big update to support pyTplot 1.4 (with xarray)	2 days ago
setup.py	going to version 0.8.3	2 days ago

README.md

pySPEDAS

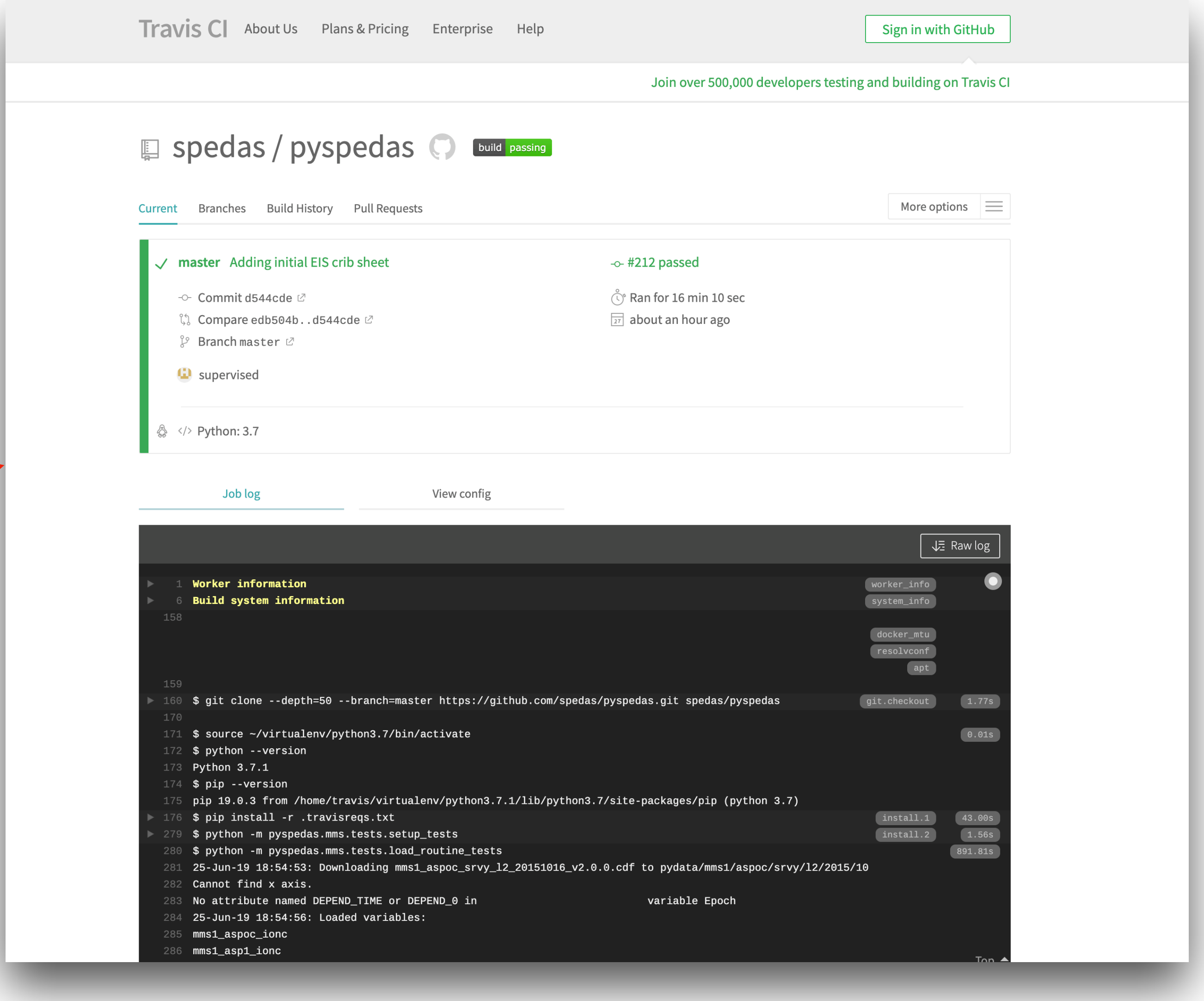
pySPEDAS is an implementation of the SPEDAS framework in python.

The Space Physics Environment Data Analysis Software (SPEDAS) framework is written in IDL and contains data loading, data analysis and data plotting tools for various scientific NASA missions.

This package is designed to work with the libraries cdflib and pytplot.

How It Works

CDF files are downloaded from the internet to the local machine. The data from these files is loaded into pytplot objects



Join over 500,000 developers testing and building on Travis CI

spedas / pyspedas

✓ master Adding initial EIS crib sheet — #212 passed

Commit d544cde
Compare edb504b..d544cde
Branch master

supervised

Ran for 16 min 10 sec
about an hour ago

Python: 3.7

```
Worker information                                                                                    worker_info
Build system information                                                                              system_info

                                                                                                       docker_mtu
                                                                                                       resolvconf
                                                                                                              apt

$ git clone --depth=50 --branch=master https://github.com/spedas/pyspedas.git spedas/pyspedas     git.checkout   1.77s

$ source ~/virtualenv/python3.7/bin/activate                                                                   0.01s
$ python --version
Python 3.7.1
$ pip --version
pip 19.0.3 from /home/travis/virtualenv/python3.7.1/lib/python3.7/site-packages/pip (python 3.7)
$ pip install -r .travisreqs.txt                                                                     install.1   43.00s
$ python -m pyspedas.mms.tests.setup_tests                                                           install.2    1.56s
$ python -m pyspedas.mms.tests.load_routine_tests                                                               891.81s
25-Jun-19 18:54:53: Downloading mms1_aspoc_srvy_l2_20151016_v2.0.0.cdf to pydata/mms1/aspoc/srvy/l2/2015/10
Cannot find x axis.
No attribute named DEPEND_TIME or DEPEND_0 in                            variable Epoch
25-Jun-19 18:54:56: Loaded variables:
mms1_aspoc_ionc
mms1_asp1_ionc
```

Current Status

- Access to team-only data (QL, L1, L2pre), as well as L2 data
- Access to support data via the `get_support_data` keyword
- Access to available data files (without downloading) with the `available` keyword
- Data can be loaded from disk without internet connectivity
- Data files are stored in temporary directories until each download completes, to avoid partial downloads of files due to internet connectivity issues
- Local data directory can be specified by the MMS_DATA_DIR environment variable, or by the 'local_data_dir' value in mms_config.py
- Local data paths match those at the SDC, as well as those used by IDL SPEDAS
- Data can be clipped to the requested time range after loading via the `time_clip` keyword
- FGM data are automatically deflagged
- EIS omni-directional spectrograms are calculated from the individual telescope data by default
- HPCA omni-directional spectrograms can be calculated
- FPI and HPCA measurements can be adjusted to the center of the accumulation interval with the `center_measurement` keyword
- Keywords for specifying instrument details (`level`, `data_rate`, `datatype`) accept strings as well as arrays of strings
- The `probe` keyword accepts strings, arrays of strings, ints and arrays of ints
- The load routines accept a wide range of time range formats via the `trange` keyword
- The load routines correctly handle access to all burst-mode data, even when small time ranges are requested
- Cross platform: tested on Windows, macOS and Linux