SSIG inputs on Time Domain

B. Cecconi, M. Louys, S. Erard

Time in Solar System Data

- **All data are time-tagged**, because most objects are dynamic (either intrinsically dynamic, like plasma, or atmospheres), or moving in space (solar illumination and observation geometry changing with time).

- Time-Series is a wide-spread (and not new) data product type in space physics:
  in-situ measurements of local medium properties with high cadence (e.g., ~1ms) over long time spans (e.g., decades!)
  => spectral analysis for periodic phenomena (waves), correlation between instruments (for transients, or waves)

- **NB**: Single spacecraft in-situ time-series, not possible to distinguish temporal from spatial variations.

- **Accurate timing and knowledge of observation geometry** at the same time is required for remote observations and subsequent analysis:
  *moving spacecraft + moving and rotation body + small field of view (i.e.: small fraction of object is observed) => **NASA/NAIF SPICE library is made for this (and very efficient)***
  => long term variations (new craters, new rings structure…) or short term variation (waves on Titan lakes, Solar spots, dust storms on Mars, Aurora on Jupiter or Saturn…)

- Not mentioning Earth observations… (Earth is a planet and is part of the Solar System :-)

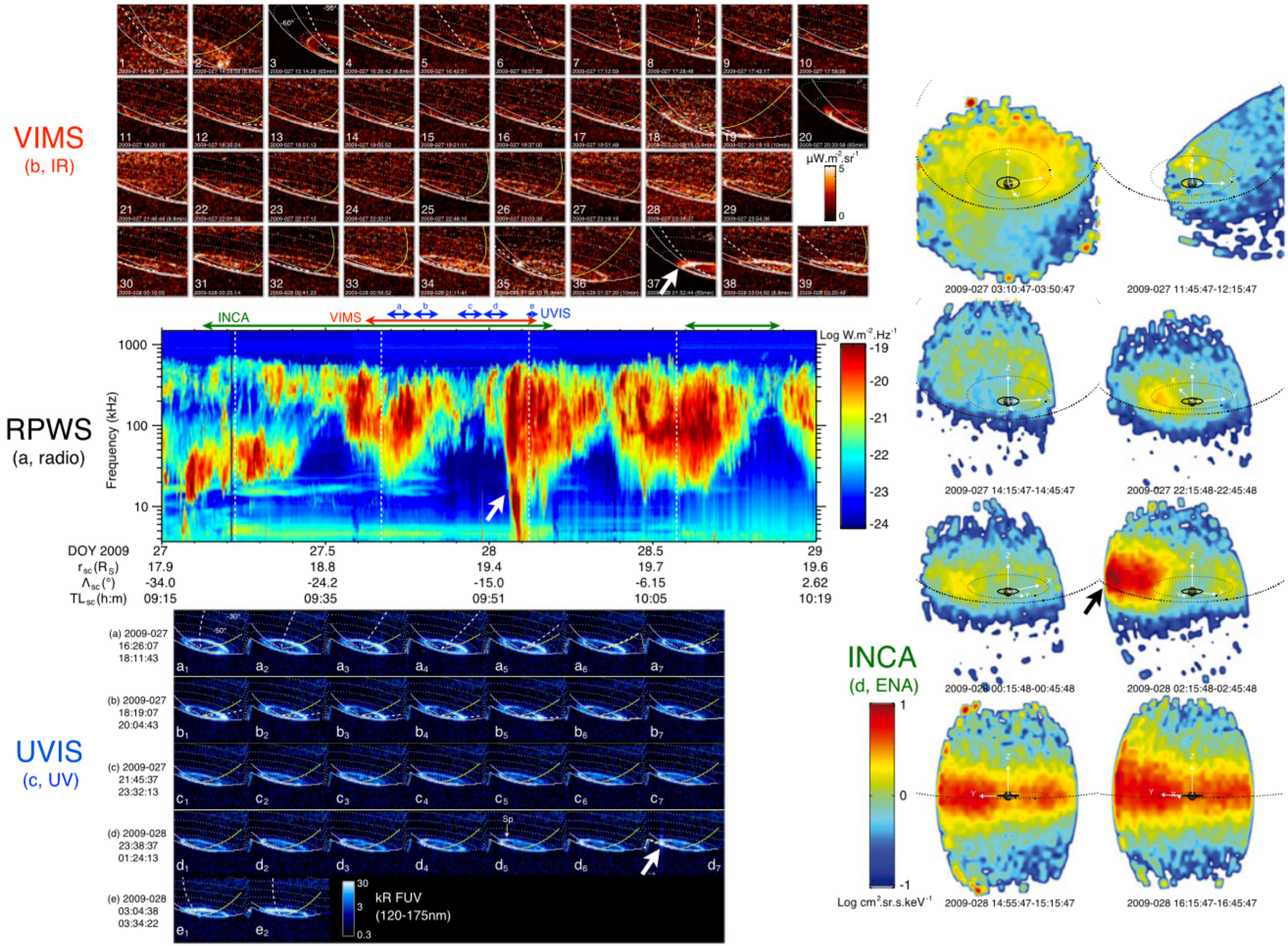
Use Cases for LF radio and plasma wave
Sampling vs Exposure vs Resolution

- **Sampling step** use case.
  1 sample every 10 ms => it’s ok for study of 10Hz-periodic signal.

- **Exposure time** use case.
  Useful for SNR determination, mixed with spectral bandwidth:
  SNR = 1. / (individual sample exposure time X sample spectral bandwidth)^0.5

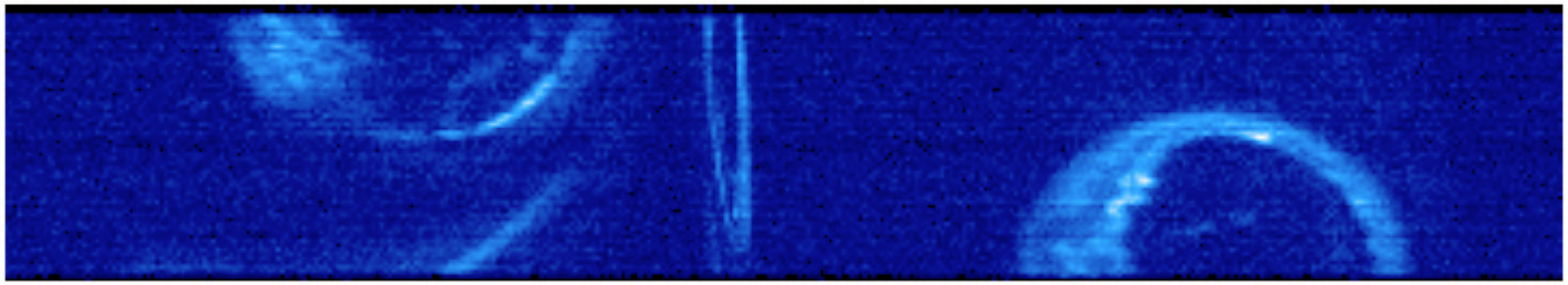
- **Resolution** use case (=dating accuracy).
  For transient events (e.g., identification of simultaneous observations of Saturn lightnings observed from Cassini and Earth), but spacecraft clock usually not accurate below 1 ms.

Example of joint observations on Cassini spacecraft: VIMS (IR) + RPWS (LF radio) + UVIS (UV)

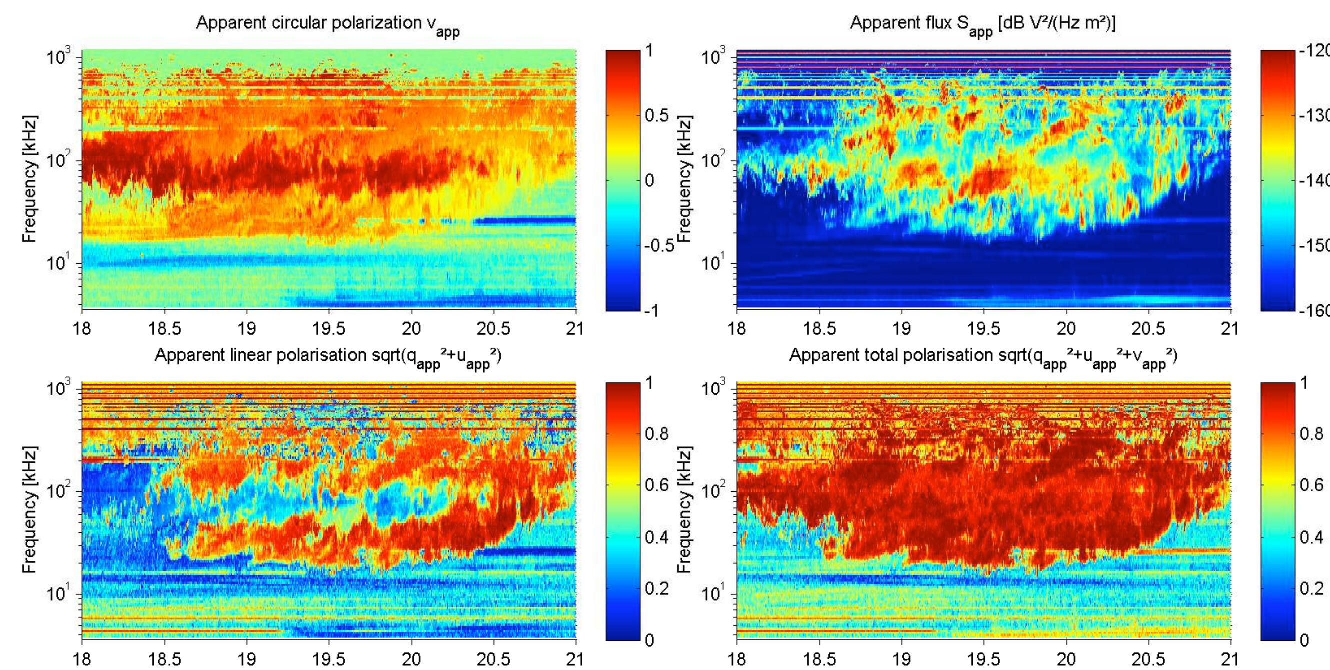


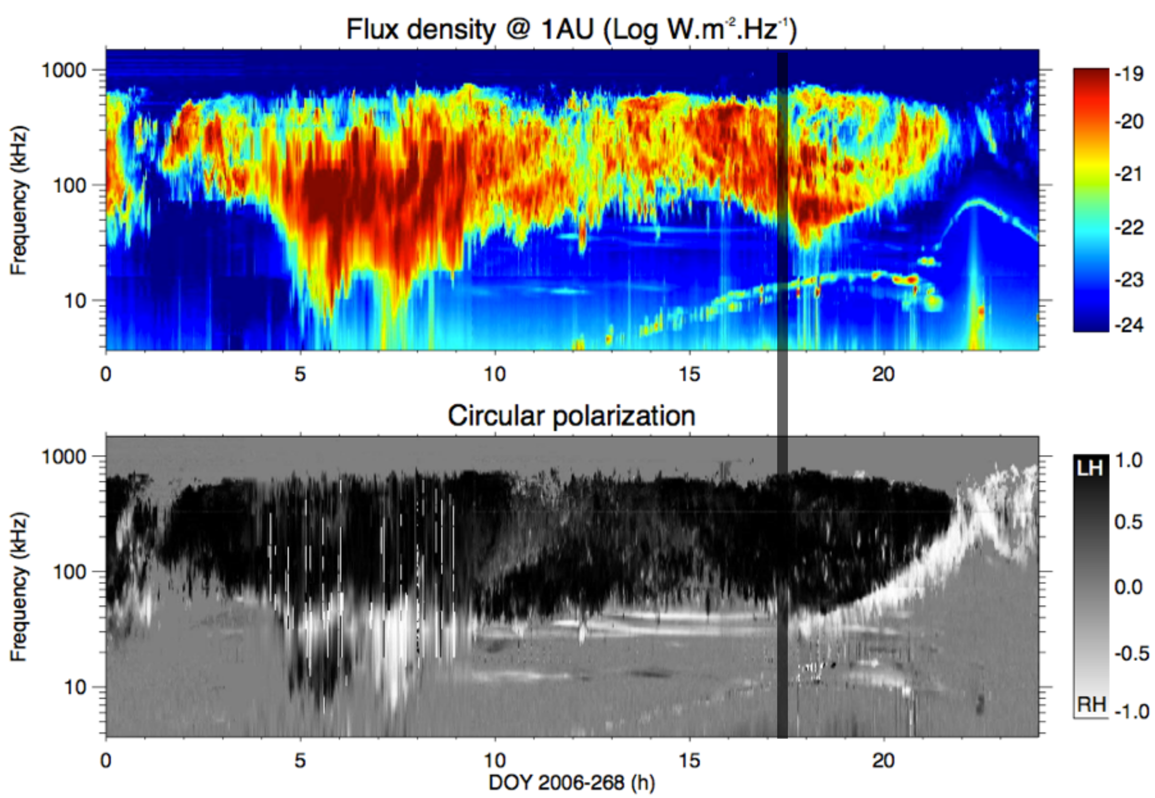
[Lamy et al, 2016]

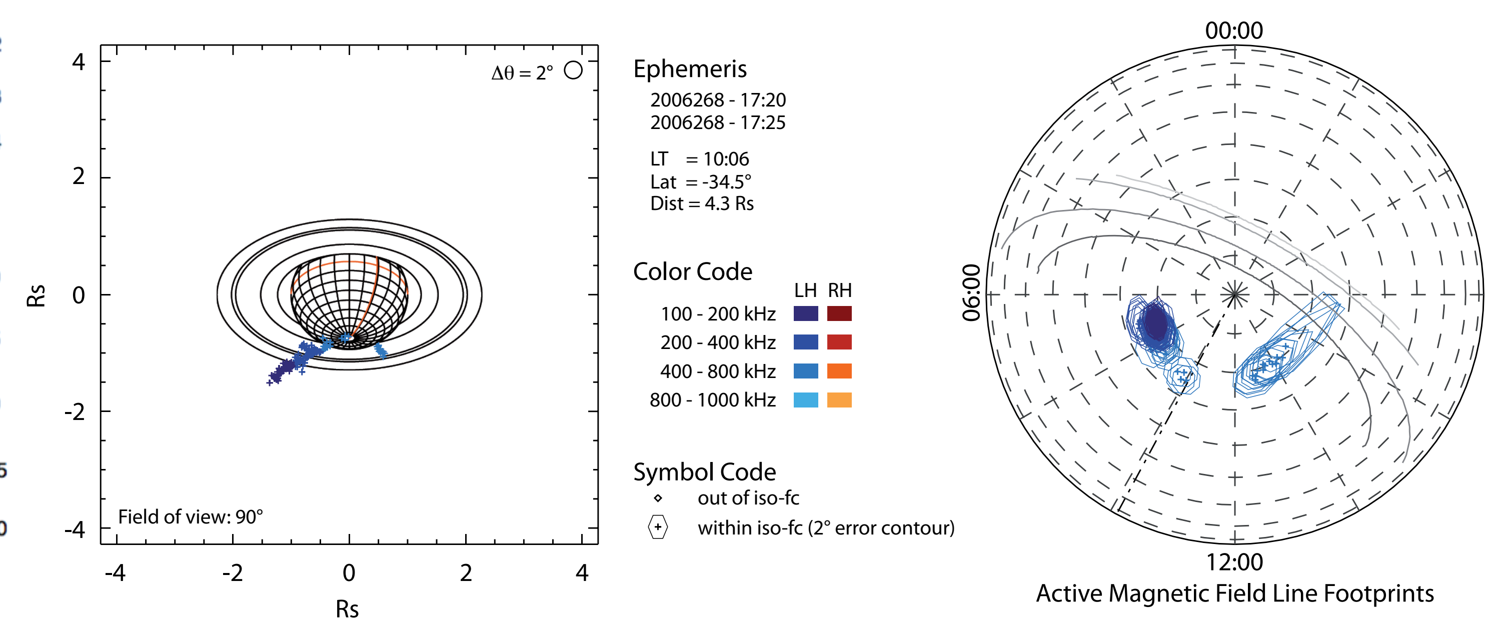
Cassini/UVIS raw data:
1 slit sweeping on target back and forth



Cassini/RPWS data:
polarization, flux, direction of arrival spectrograms

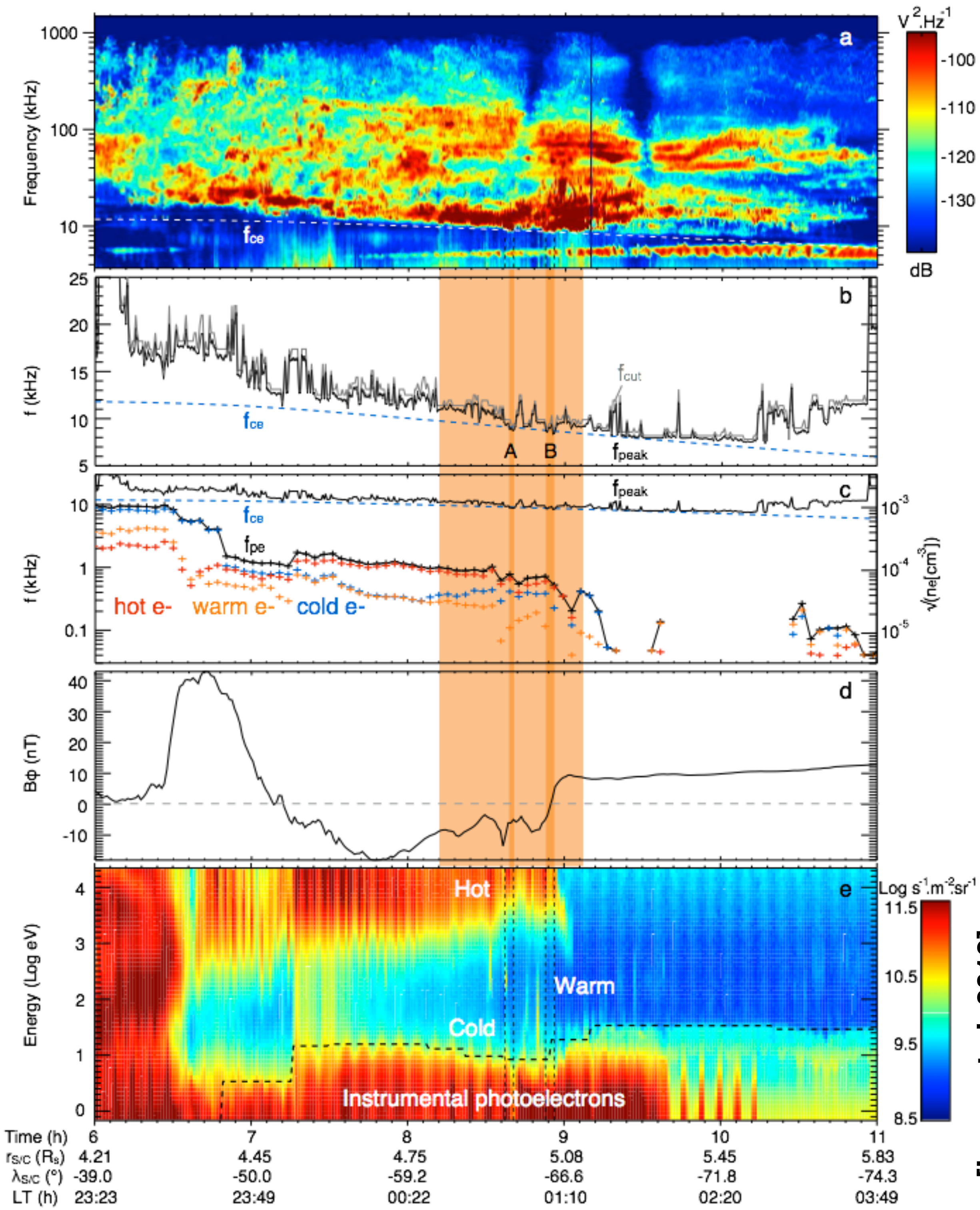




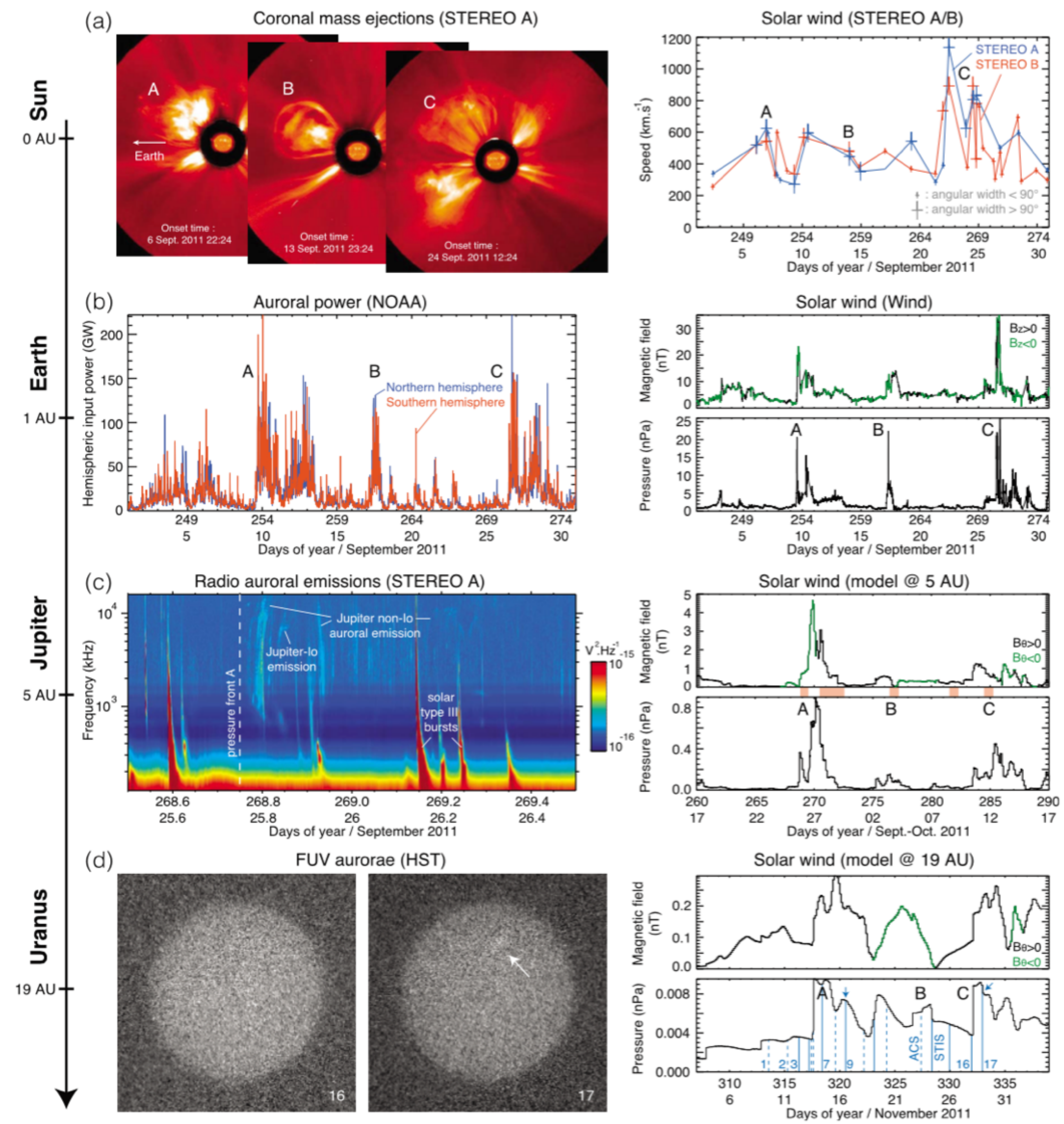


Remote Sensing + In situ observations of auroral radio source at Saturn (Cassini/MAG + Cassini/CAPS + Cassini/RPWS)

Tracking Interplanetary shock from the Sun, to Earth, Jupiter and Uranus, modeled propagation compared with auroral phenomena

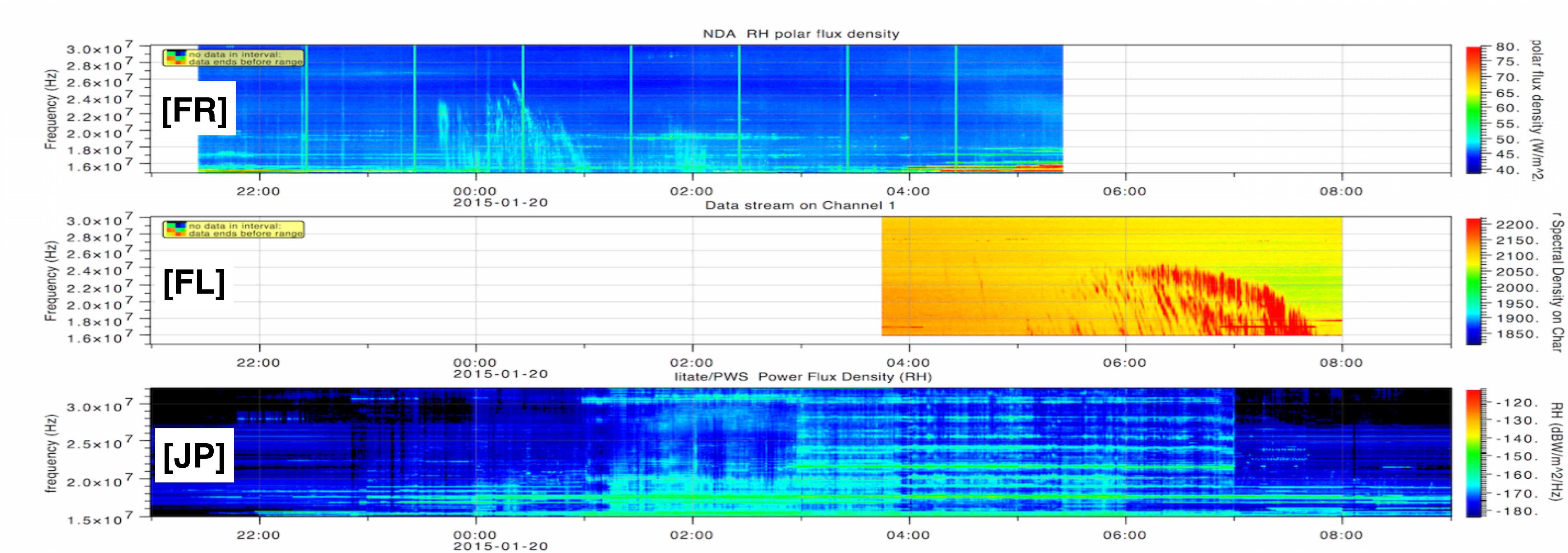


[Lamy et al, 2012]

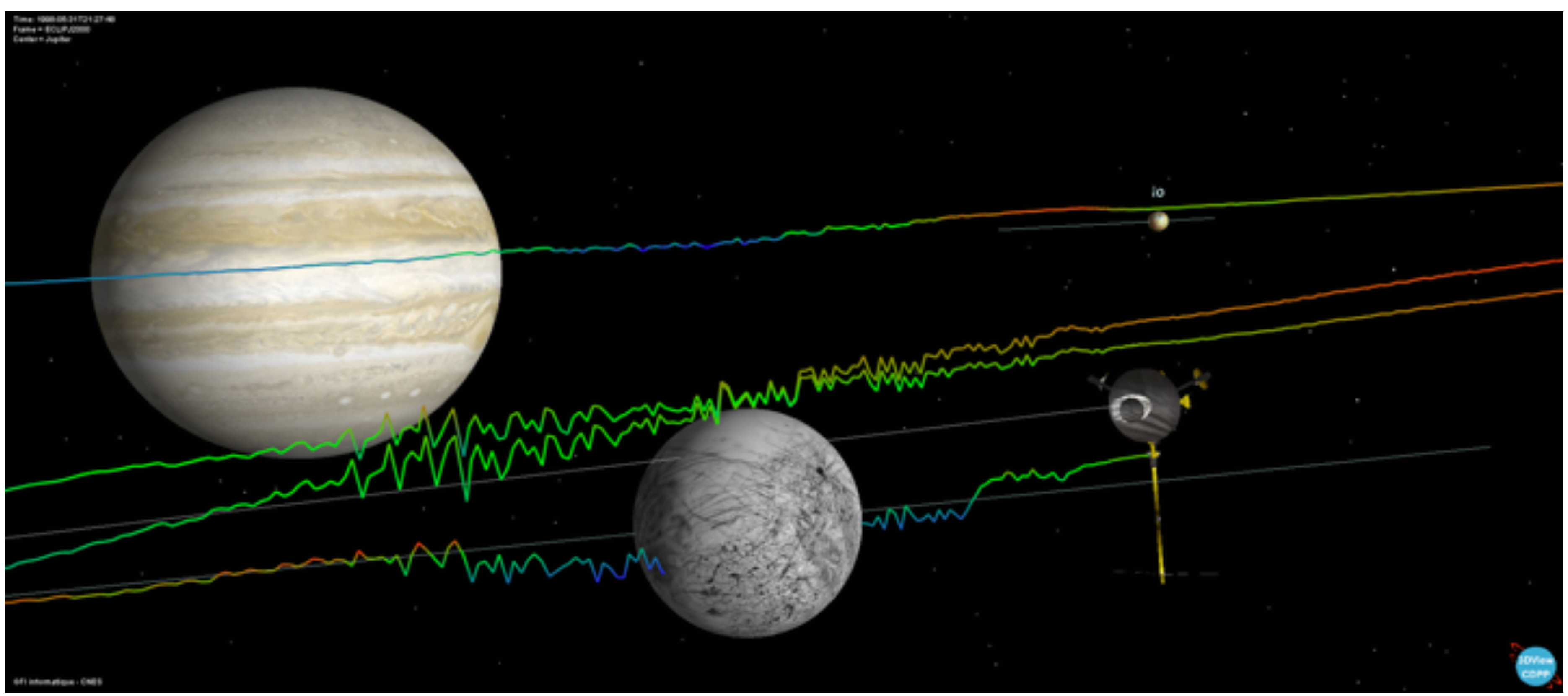


[Lamy et al, 2013]

Juno Ground Radio LF support (with Autoplot)

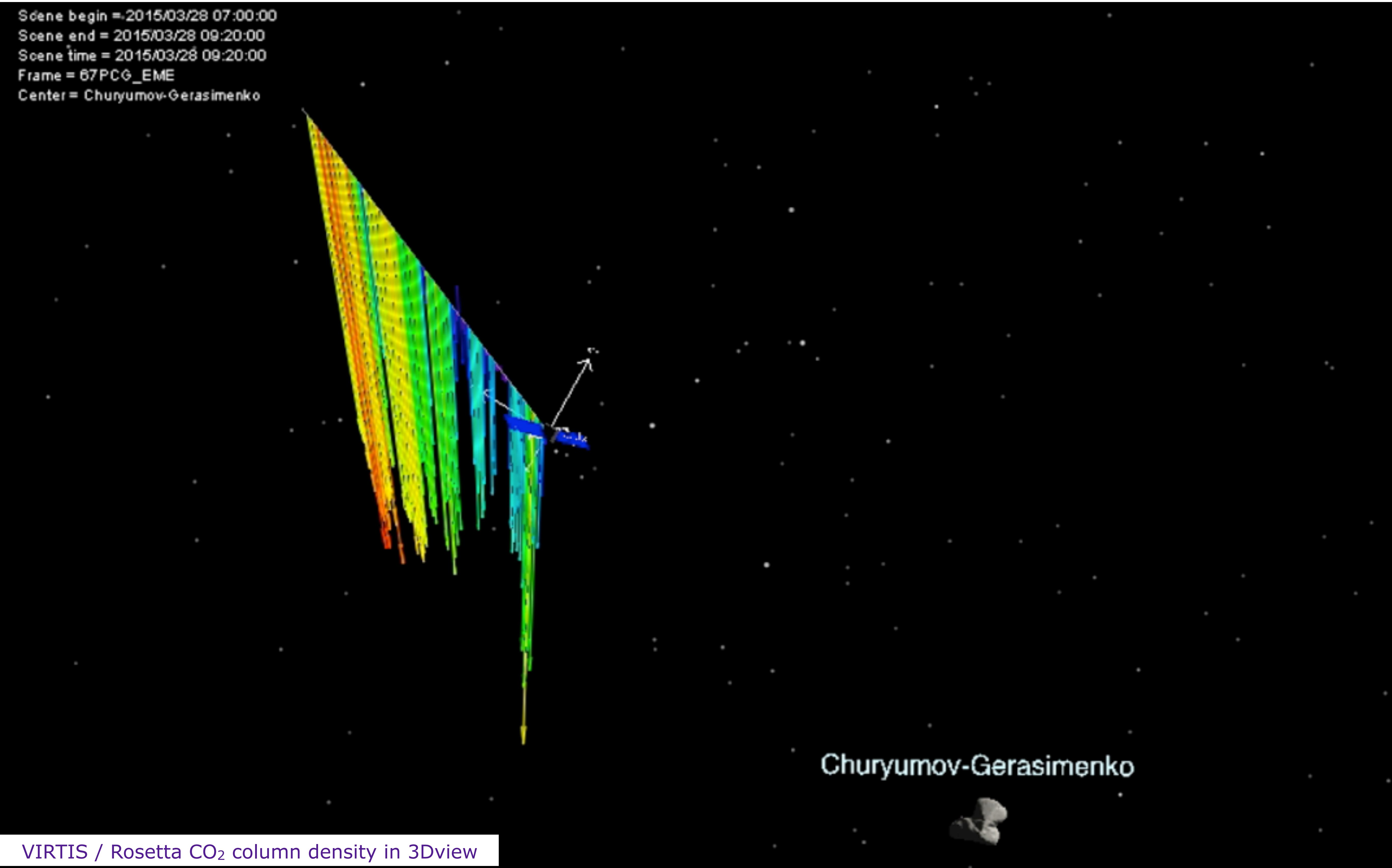


Galileo Flyby of Europa (MAG instrument, with 3Dview)



Rosetta @ CG-67P **use case:**

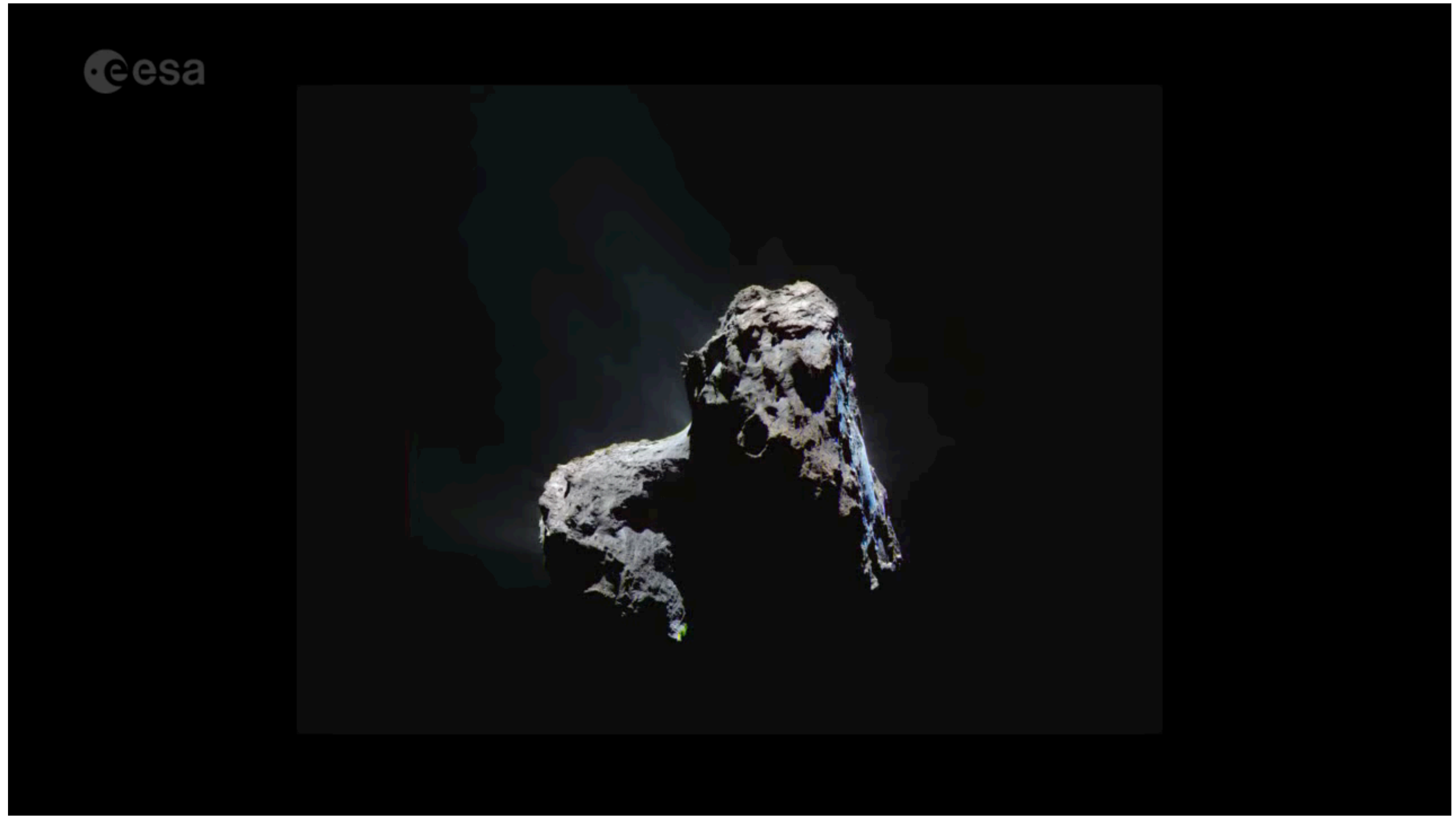
Assessment of use as a 3D viewer for atmospheric / coma observations (using e-m field plotting mode)



VIRTIS / Rosetta CO_2 column density in 3Dview

Rosetta @ CG-67P

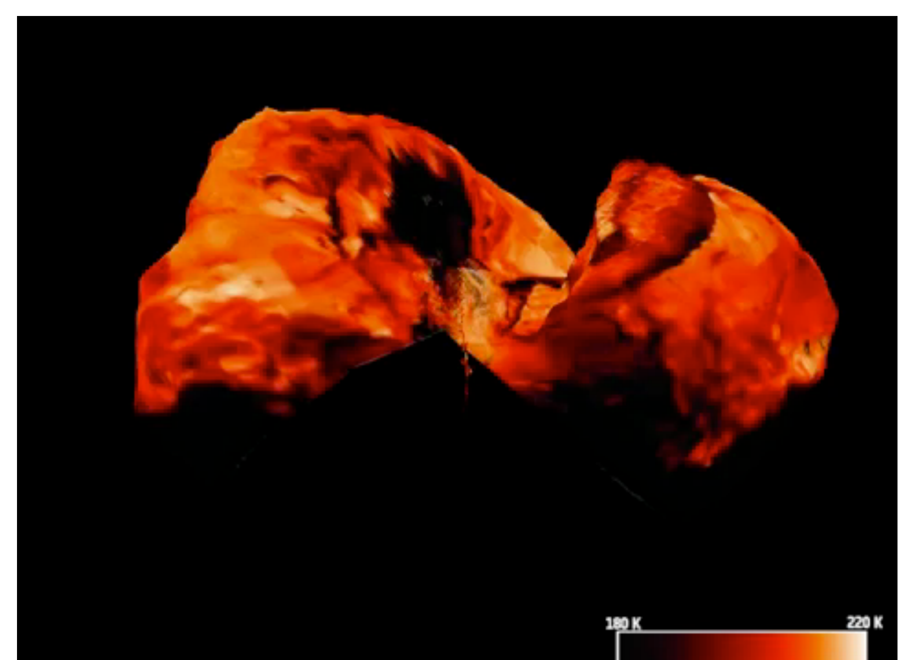
time varying modeled illumination map



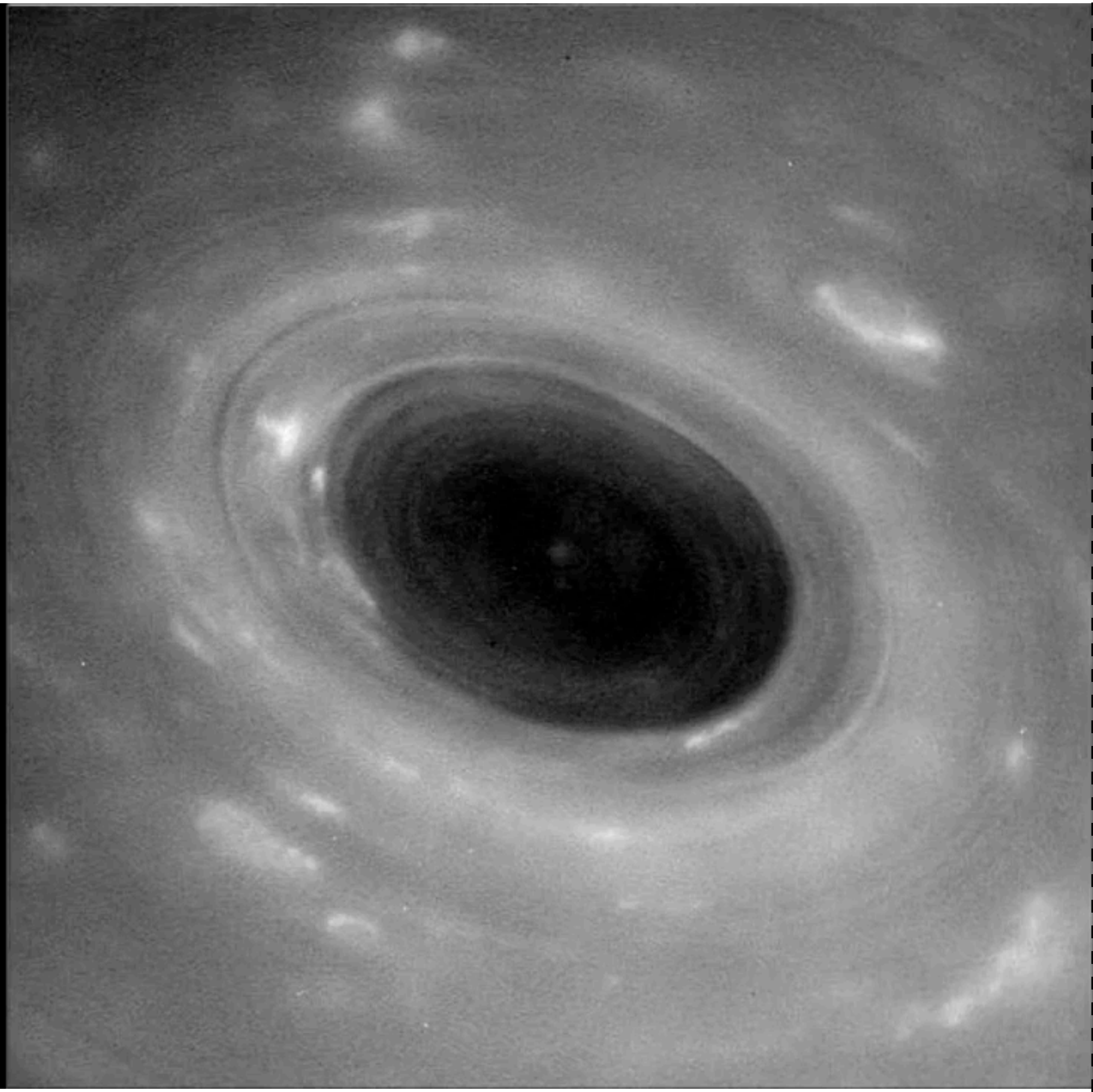
composite images of various instruments

Imaging/spectroscopy
+ illumination conditions
(both time-varying)
to build a heat map =>

67P Churyumov-Gerasimenko
VIRTIS heat map on Shapemodel

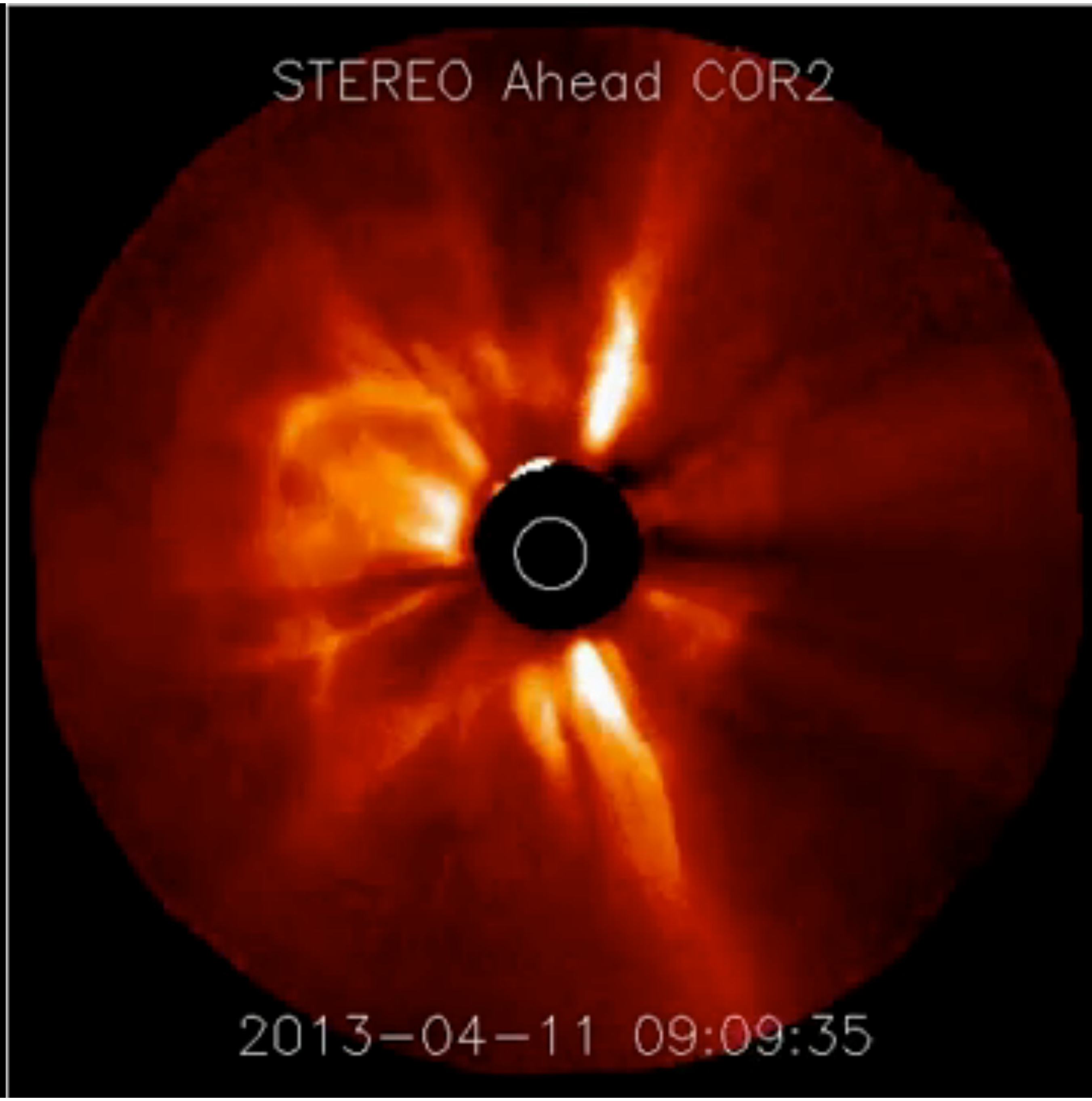


Cassini first “proximal” orbit
(without time or pointing context)



impossible to interpret
without accurate space-time knowledge

STEREO-A CME
spacecraft in the path



time tagged images
with remote and local signals

Time scales / formats
SCLK vs SCET vs UTC

- Spacecraft clock (SCLK) is internal Spacecraft clock, drifting + jitter.
  It is counting the number of clock ticks since beginning of mission (or clock restart).

- SCLK is used to compute Spacecraft Event Time (SCET) by Operation center of Space Agency

- SCET = UTC @ spacecraft, corrected for one way light-travel time.

- SCET is usually the time used for data analysis,
  except for transient identification between instruments on the same Spacecraft: in this case use SCLK.

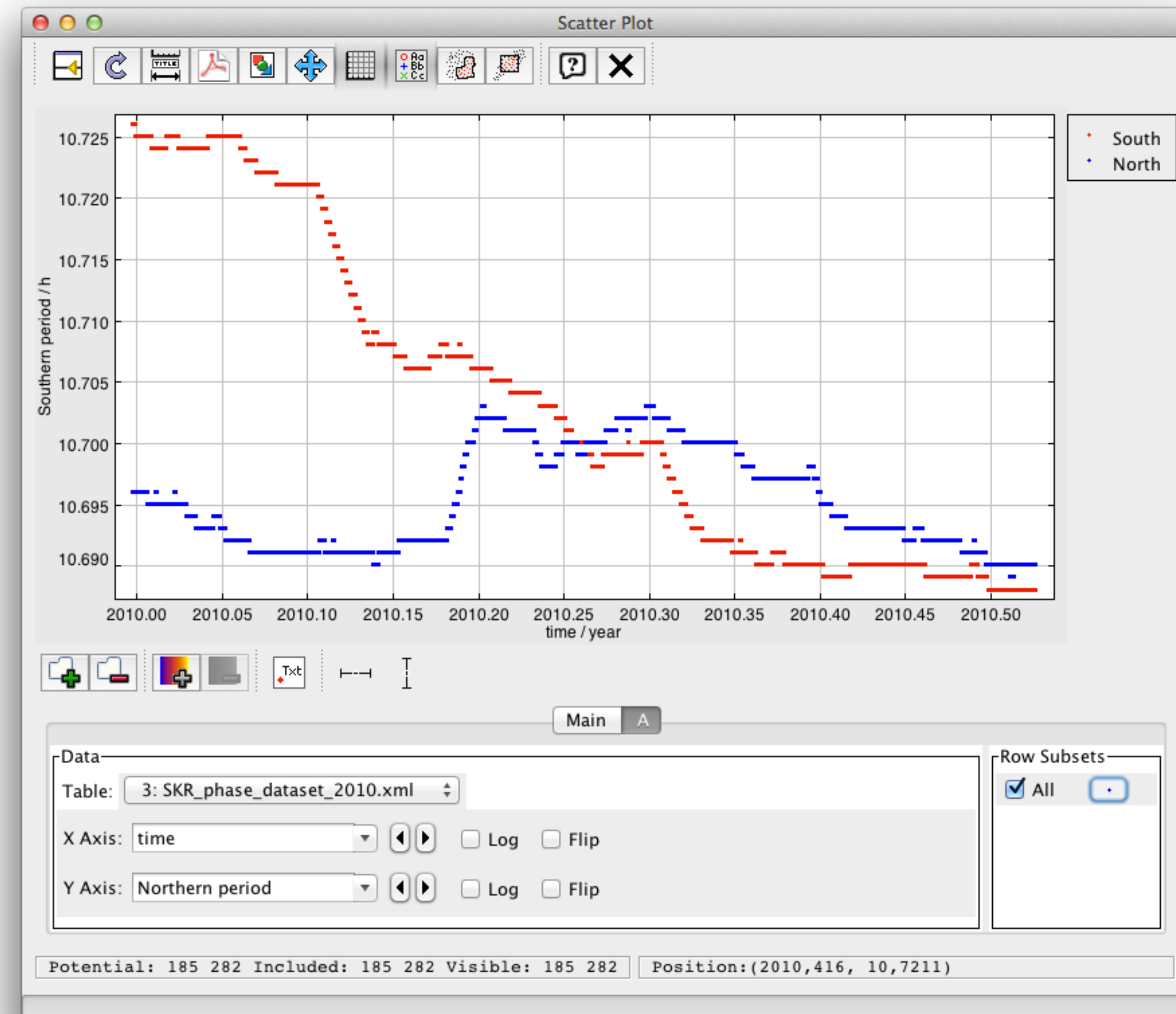
- Sometimes, TT2000 is used as in CDF-A (archive version).

- Times are provided in various formats or units:
  - mostly ISO-time (most software libraries or database handle this smoothly)
  - decimal day from given Epoch (could be JD, but not only)
  - elapsed seconds (or nanoseconds) from a given Epoch…

Other temporal axes

- In addition to UTC and alike, temporal axes **related to either planetary sidereal rotation or orbital period**:
  - planetary diurnal time (e.g., based on Martian day)
  - planetary season (solar longitude of Mars on its orbit)
  - solar rotation (decimal Carrington rotation number)

- For giant planets, longitude system is not unique => **several planetary rotation** temporal axis.
  - Jupiter: 3 different longitude systems => 3 reference rotation periods (1 for core rotation, 2 for cloud systems) => 3 temporal axes
  - Saturn is even more complex: true sidereal period is unknown. Reference period in literature is debated => many longitude systems (even variable period longitude systems, depending on hemisphere!)



Data discovery or catalog mining EpnCore temporal queries

- Using temporal parameters for data product selection or discovery.

- Criteria:
  - Search for data including a given epoch or interval
  - Search for data included in a given interval
  - Search for data with a specified sampling step condition (larger or small than a given threshold)
  - Search for data with a specified exposure time [or integration time] condition (larger or small than a given threshold)
  - Select/check on the time scale and/or time origin
  - Search for data with a time condition at observer
  - Search for data with a time condition at target
  - Search for data with “local time” or “season”

EPNcore/ObsCore
Temporal time interval

Data Model	ObsCore	EpnCore
keyword	**t_min** **t_max**	**time_min** **time_max**
description	Start/Stop time	Acquisition start/stop date-time or date-time of the first/last sample
Unit	MJD	JD (ISO under study)
UCD	time.start;obs.exposure time.end;obs.exposure	time.start time.end
utype	*Char.TimeAxis.Coverage.Bounds.Limits.StartTime* *Char.TimeAxis.Coverage.Bounds.Limits.StopTime*	

EPNcore/ObsCore Temporal sampling step

Data Model	ObsCore	EpnCore
keyword	–	**time_sampling_step_min** **time_sampling_step_max**
description	—	Minimum/maximum value of the temporal sampling step (minimum/maximum value of the time interval between two successive individual acquisitions)
Unit	—	s
UCD	—	time.interval;stat.min time.interval;stat.max
utype	*Char.TimeAxis.SamplingPrecision.SamplingBounds.Period.Limits.LoLimit* *Char.TimeAxis.SamplingPrecision.SamplingBounds.Period.Limits.HiLimit*	

EPNcore/ObsCore Observation exposure time

Data Model	ObsCore	EpnCore
keyword	**t_exptime**	**time_exp_min** **time_exp_max**
description	Total exposure time	Minimum/maximum value of the individual integration/acquisition/exposure time in data product
Unit	s	s
UCD	time.duration;obs.exposure	time.duration;obs.exposure;stat.min time.duration;obs.exposure;stat.max
utype	Char.TimeAxis.Coverage.Support.Extent	*Char.TimeAxis.SamplingPrecision.SamplingBounds.Extent.Limits.LoLimit* *Char.TimeAxis.SamplingPrecision.SamplingBounds.Extent.Limits.HiLimit*

In ObsCore, t_exptime is the integrated exposure time for the data product (e.g. when the exposure time of distributed image or spectrum is the sum of individual snapshots), so it is a different concept than the time_exp_min/time_exp_max in EPNcore. We may need to add such concept in EPNcore. It would yet another column, but that could be useful, and it would help convergence with ObsCore.

EPNcore/ObsCore Observation time resolution

Data Model	ObsCore	EpnCore
keyword	**t_resolution**	**(time_exp_min?)**
description	minimal interpretable interval between two points along the time axis	—
Unit	s	—
UCD	time.resolution	—
utype	*Char.TimeAxis.Resolution.ResolutionRefVal.ReferenceValue*	

Data products with time information

- Time-series = a table (whatever file format), with a time value and a series of parameter values/vectors/matrices for each time step.

- Spectrograms (a.k.a., dynamic spectra): property measured with temporal and spectral dependence (either from FFT of high resolution time-series, or from a series of filters sampled at each time step)

- (Spectral)cubes: either series of spectra/images measured at different times, or within a signal spectrum/image, each pixel/bin may not be measured at the same time (sweeping spectrometer), and thus not exactly sampling the same location on target (moving observer and target!).

- Events: observation or prediction.
  VOEvent description: <wherewhen> = STC

Data product content

- In time-series data product:
  - Temporal sampling step is intrinsically present if you have successive samples.
  - Temporal resolution/exposure time needs to be added as an extra column (if sample dependent) or metadata (same value for all sample).
  - All instrumental parameters (spectral, pointing, polarization…) can be time-dependent (changing at each sample):
    => **often sparse multi-dimensional data with main axis = time**

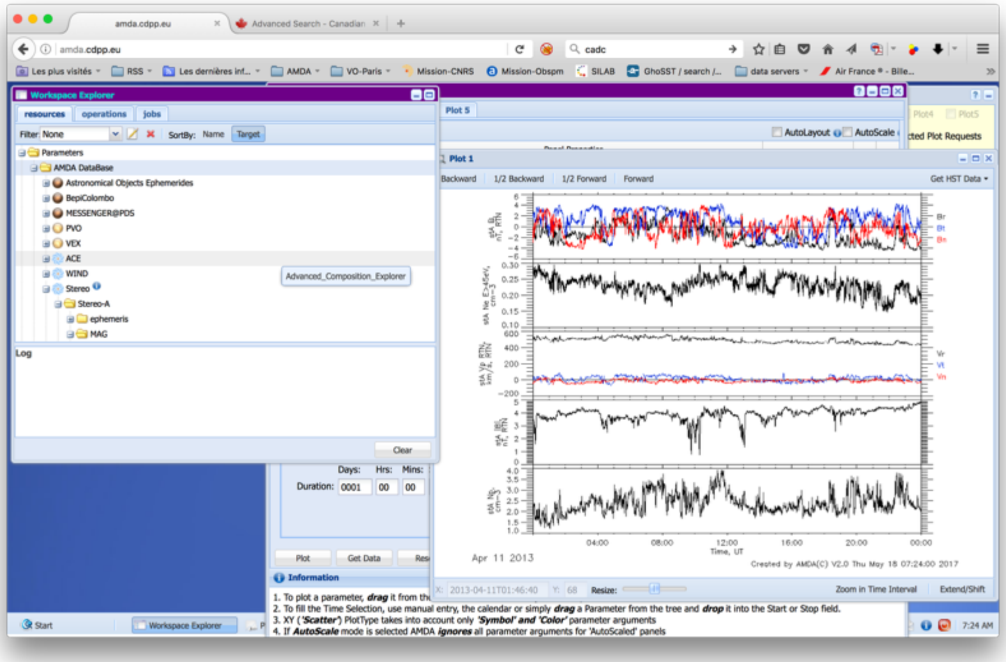
- Publishing time-series as files:
  - as a VOTable: 1 row = 1 sample. (1 column for time stamps + other columns…),
  - as FITS (e.g., a series of cubes with within a time stamp for each image or spectra),
  - as CDF (time stamp is mandatory), etc.

  Not sure why an extra file format would be needed.

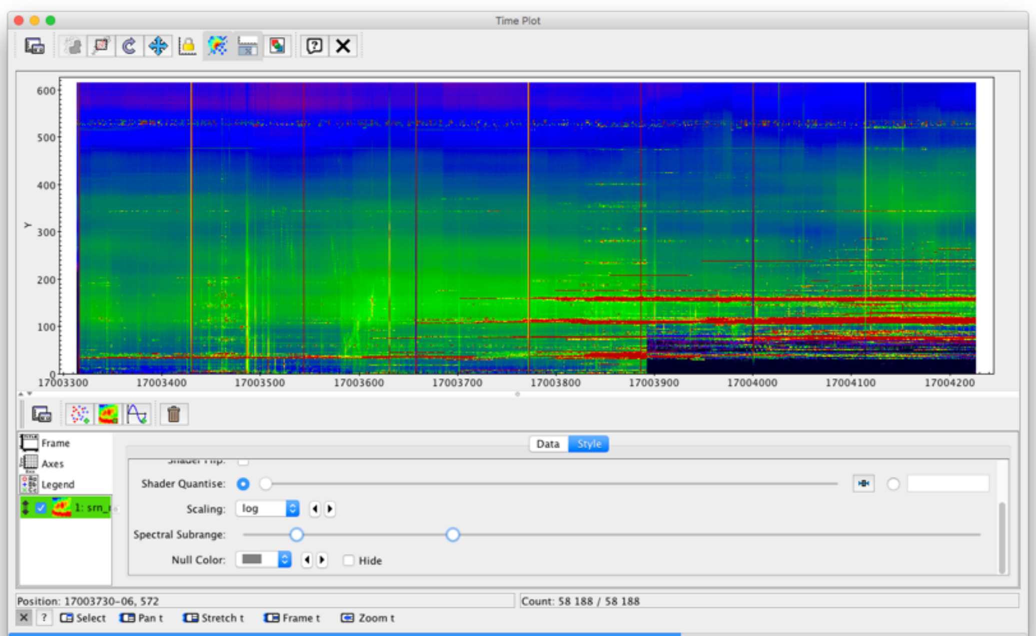
- Mandatory File+Search metadata:
  - time scale (+starting point/epoch), time origin (location)
  - time coverage + sampling/resolution/exposure

Existing display tools

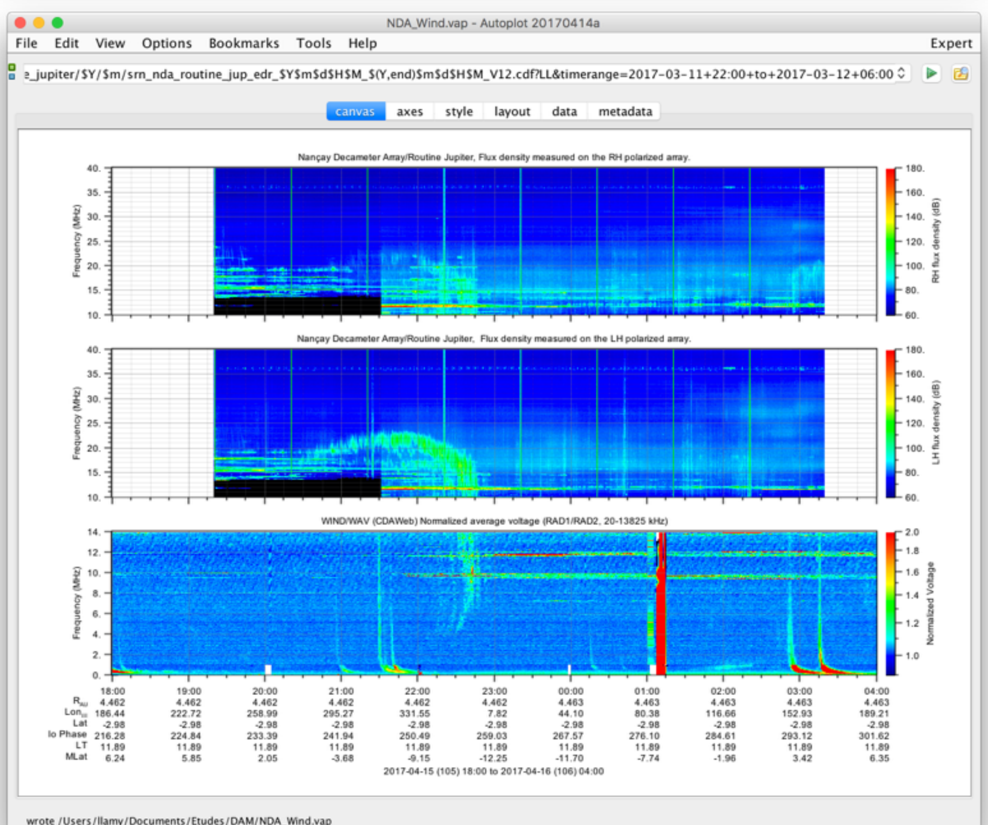
- AMDA (CDPP, Toulouse, France)
  http://amda.cdpp.eu



- TOPCAT (Mark Taylor, Bristol, UK)
  Time Plot option



- Autoplot (Univ. Iowa, USA)
  http://autoplot.org
  (SAMP connexion now possible)



Temporal analysis

- Many analysis methods exist for analyse temporal data:
  - FFT (periodic, regularly sampled)
  - periodograms (periodic, regularly sampled)
  - lomb scargle (uneven and sparse sampling)
  - Hilbert-Huang (decomposition in component with varying periodicities)
  - scalograms/wavelet transforms (variabilities depending on scales): huge numbers of wavelet types
  - chirplets (periodic drifting tones)
  - …

- Complete data processing suite ?
  => python, IDL, matlab ? :-)

Time-series data service ?

- Large data set (long-lived mission with high resolution sampling) => difficult to access long time interval (for display or statistical analysis)

- NASA/Heliophysics is working on an API for serving time series [**Heliophysics API = HAPI**]. Lightweight protocol.

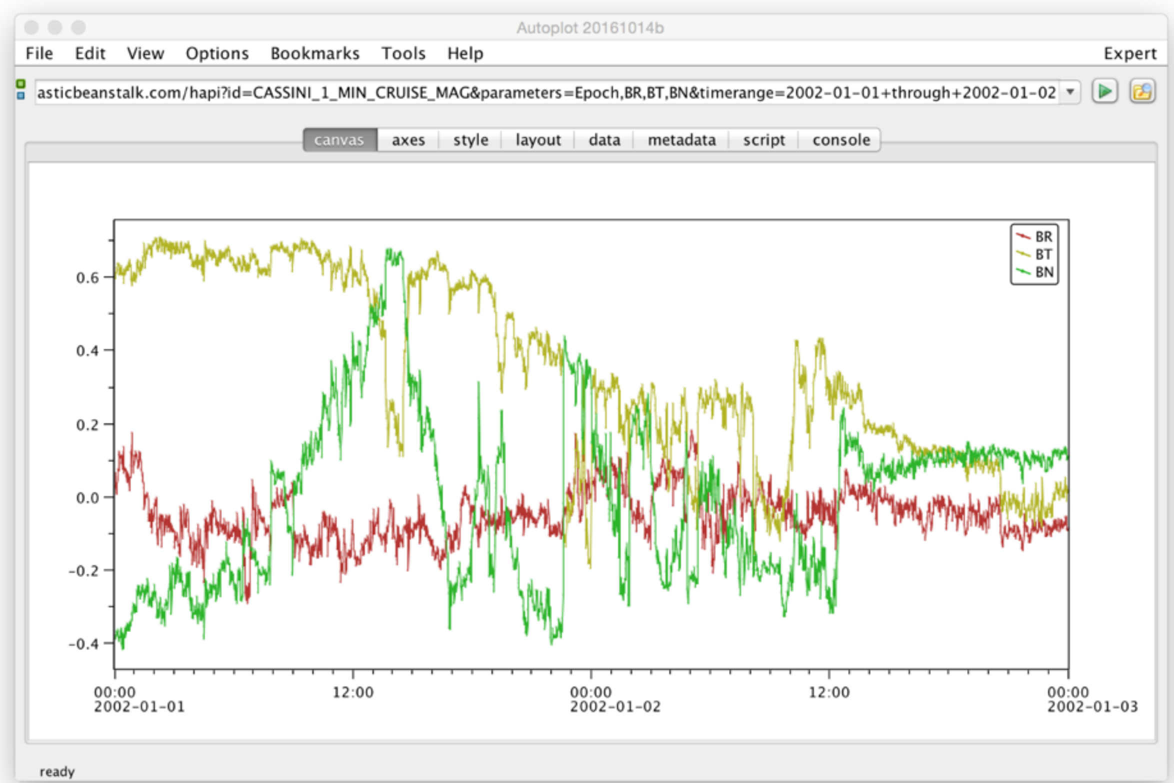
http://example.com/hapi/capabilities

- response is in JSON format
- all outputs include the API version number
- indicates which optional elements of the serer are implemented by this server
- currently, the only option is the output formats of the data

  which are CSV (required) and binary (optional) – more on this later
- Example:

```
{
   "HAPI" : "1.0",
   "capabilities" :
   [
      {
         "formats": ["csv", "binary" ]
      }
   ]
}
```

vap+hapi:http://datashop.elasticbeanstalk.com/hapi?id=CASSINI_1_MIN_CRUISE_MAG&parameters=Epoch,BR,BT,BN&timerange=2002-01-01+through+2002-01-02



Time-Series Data
Where is the complexity?

- Not sure if it is so complex! 😅

- Even if different approaches:
  - Astronomy : times series = series of individual images not measured in a row
  - Solar System : times series = intrinsic organization of raw data
  => in both cases, observational parameters can change from a sample to the other.

- Complexity is mostly on making sure the same time axis / time scales and references are used (at the accuracy required by your science case, so it may not be very restrictive), or that they can be translated one to the other, with correct description.
  => best is to use UTC @ specified location with non ambiguous time unit/format (MJD not recommended?)

- Solar system sciences is using time as the main axis, huge experience and knowledge.

- As far as I can tell:
  Data formats and data models are already available and seems operative