Inter-comparison of atmospheric data using the

BASIC **ENVISAT** ATMOSPHERIC TOOLBOX

science [&] technology

Overview

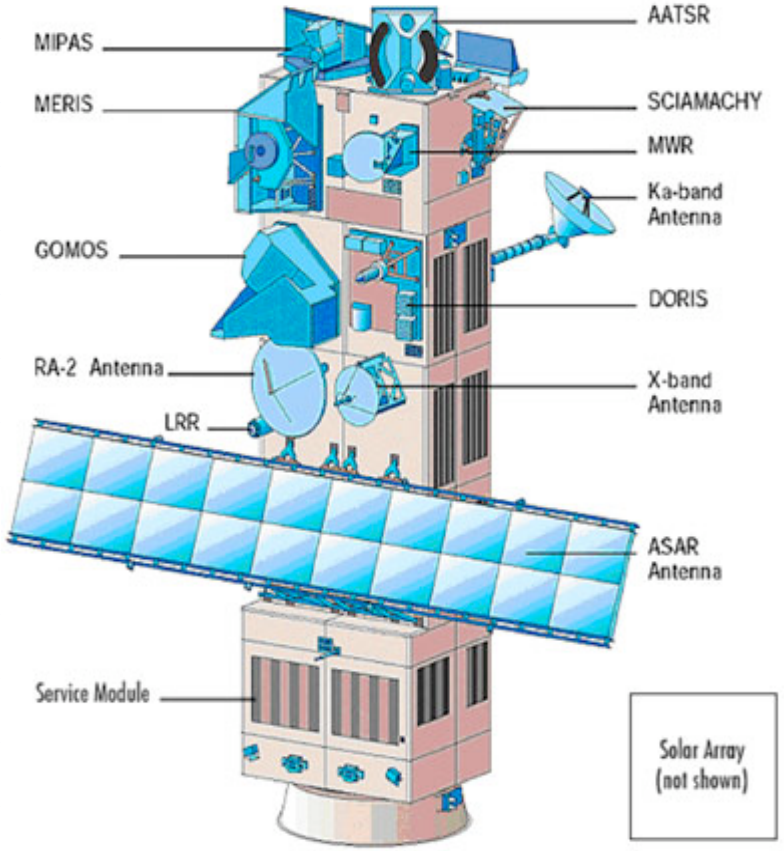
- The Envisat satellite: instruments, data products, and toolboxes
- Short overview of BEAT
- Ingestion and inter-comparison with BEAT
- Some examples

Envisat satellite - launch

- ESA satellite
- Launched on 28 February 2002
- Successfully brought into orbit

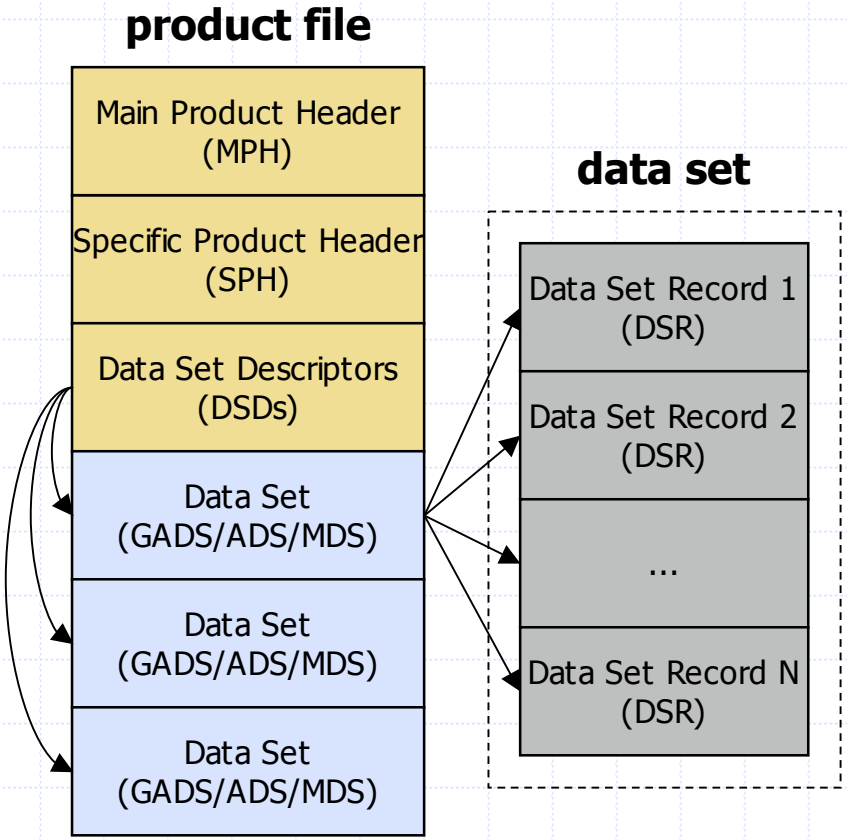
Envisat satellite - instruments

- 10 instruments
  - AATSR
  - MERIS
  - ASAR
  - GOMOS
  - MIPAS
  - SCIAMACHY
  - MWR
  - DORIS
  - RA-2
  - LRR



Envisat satellite - product data

- New common data format for all Envisat data
- A single file is a combination of ASCII headers and binary data sets
- Fixed product formats
- Not self-describing



Envisat satellite - toolboxes

- Traditionally, scientific users write their own access routines for data processing:
  - Envisat data files can be quite complex, so writing reading routines can be very time consuming and error prone
  - big waste of effort
  - scientists are doing a software engineer's job
  - same goes for some basic analysis and visualisation routines

Envisat satellite - toolboxes

- Three open source toolboxes:
  - **BEAT** - GOMOS, MIPAS, and SCIAMACHY (and GOME)
  - **BEAM** - (A)ATSR and MERIS
  - **BEST** - (A)SAR
- Different toolboxes because of added visualisation and analysis capabilities
- All three toolboxes are funded by ESA, freely available and open source

BEAT

BEAT: Basic Envisat Atmospheric Toolbox

- Toolbox for atmospheric instruments GOMOS, MIPAS, SCIAMACHY, and GOME:
  - Provide read/ingestion functionality
  - C libraries + command line tools
  - interfaces for MATLAB, IDL, and Fortran
  - includes cross platform GUI application for visualisation and analysis
  - 'supports' instrument inter-comparison

BEAT - Three layers

- BEAT-I: direct product interface (like HDF)
  - ingest each and every byte of data in a product-file to C, Fortran, IDL or MATLAB
- BEAT-II: abstract interface
  - support routines for easier ingestion, handling large volumes of data (multiple product files), filtering, and instrument inter-comparison
- VISAN:
  - GUI application for visualisation and analysis

BEAT-II

- Only one call needed to ingest the *most important* data from a single product or multiple products
- Powerful ingestion filter options to place restrictions on time, location, etc.
- Uses BEAT-I (DPI) for reading of Envisat products, but also supports reading data stored in other formats, such as HDF
- Generic 'record' type for returning data

BEAT-II - types of data

- Different kinds of atmospheric data per product level:
  - level 1:
    - Spectral readouts
    - Reference spectra
  - level 2:
    - Total column data
    - Profile data
  - level 3/4:
    - Global geo maps

BEAT-II records

- BEAT-II supports inter-comparison of instruments by solving the structural differences for similar typed data
- Standardised record types for e.g.
  - spectral readout data
  - ground pixel data
  - profile data
  - geo map data
- A record can be extended with instrument specific fields

BEAT-II record example

basic 'ground pixel' record:

type	string (e.g. 'SCI_NL__2P_geo_pixel')
species	string (e.g. 'o3')
time	double[measurements]
corner_latitude	double[measurements,4]
corner_longitude	double[measurements,4]
latitude	double[measurements]
longitude	double[measurements]
value	double[measurements]
value_unit	string (e.g. 'DU')
error	double[measurements]
error_unit	string (e.g. '%')

BEAT-II - MATLAB example

```
>> files = dir('*.lv2'); filenames = strvcat(files.name);
>> data = beatl2_ingest(filenames,'time_min=30-SEP-2000,
                        time_max=01-OCT-2000,backward_scan=0')

data =
                      type: 'GOME_L2_ground_pixel'
                   species: 'o3'
                      time: [19239x1 double]
           corner_latitude: [19239x4 double]
          corner_longitude: [19239x4 double]
                  latitude: [19239x1 double]
                 longitude: [19239x1 double]
                     value: [19239x1 double]
                value_unit: 'DU'
                     error: [19239x1 double]
                error_unit: '%'
      solar_zenith_angle: [19239x1 double]
        los_zenith_angle: [19239x1 double]
  relative_azimuth_angle: [19239x1 double]
          subset_counter: [19239x1 int32]
```

What does an ingestion do?

- provides mapping to generic record (description documented)
- mapping can be complex and require processing (e.g. co-adding geolocation data that is in a different resolution)
- unit conversion
- filtering data (combining multiple files)

BEAT-II - advantages

- No need to know product format
- Standardised nomenclature
- Standardised units (where appropriate)
- Write generic (visualisation) algorithms
- Records are easy to modify/extend
- Easy comparison of similar typed data (i.e. instrument inter-comparison)

VISAN

The advantages of having standardised records have been applied to VISAN:

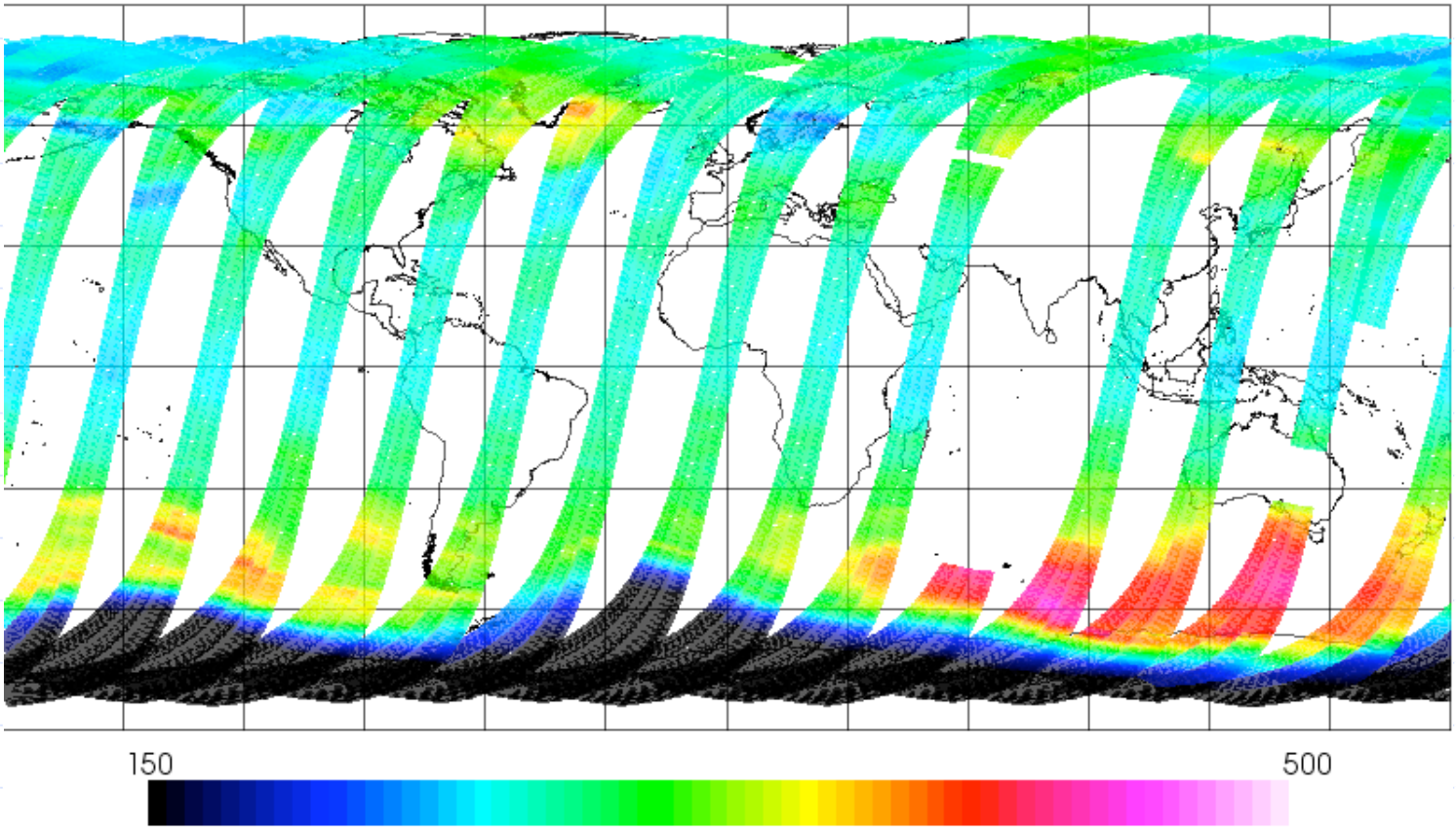
- Provide a 'one call' visualisation method:

```
>>> data = beatl2.ingest(<files>,'<filter>')
>>> plot(data)
```

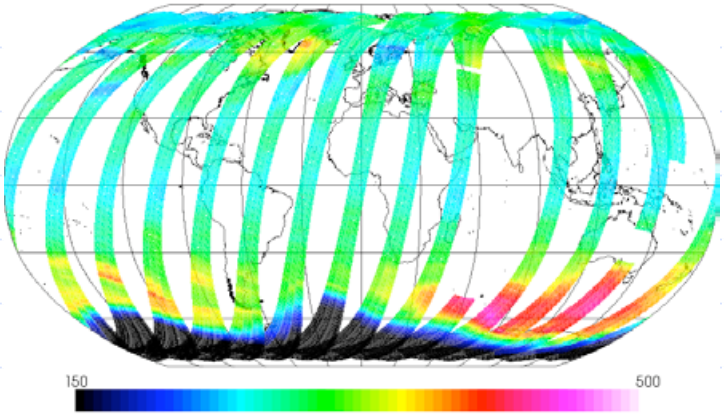
or

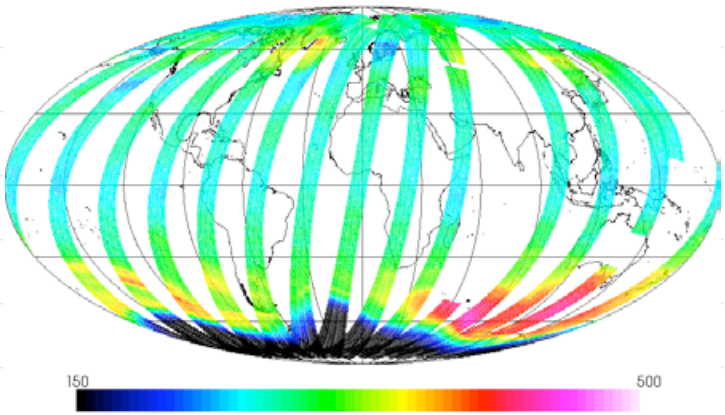
```
>>> wplot(data)
```

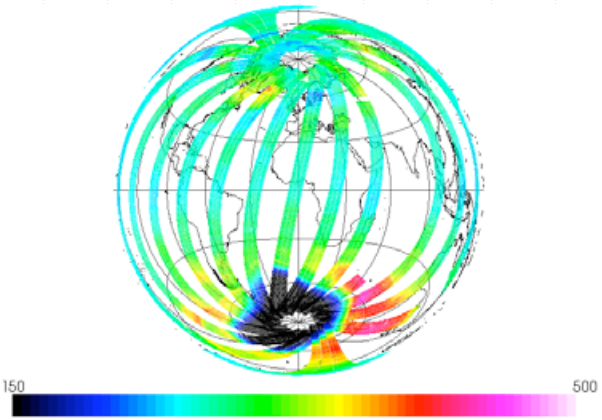
VISAN - GOME L2 example

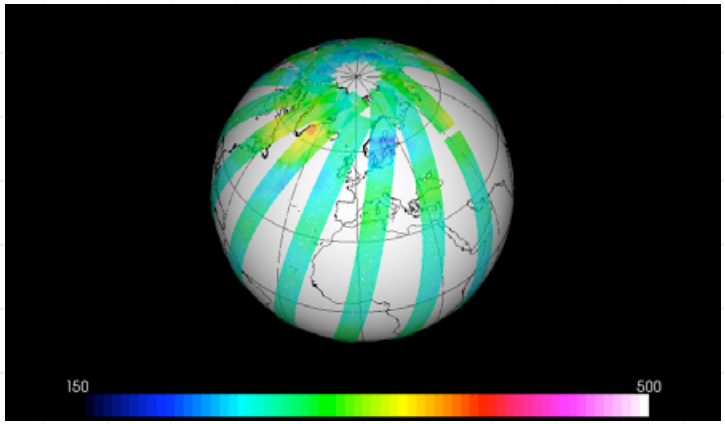


VISAN - different projections

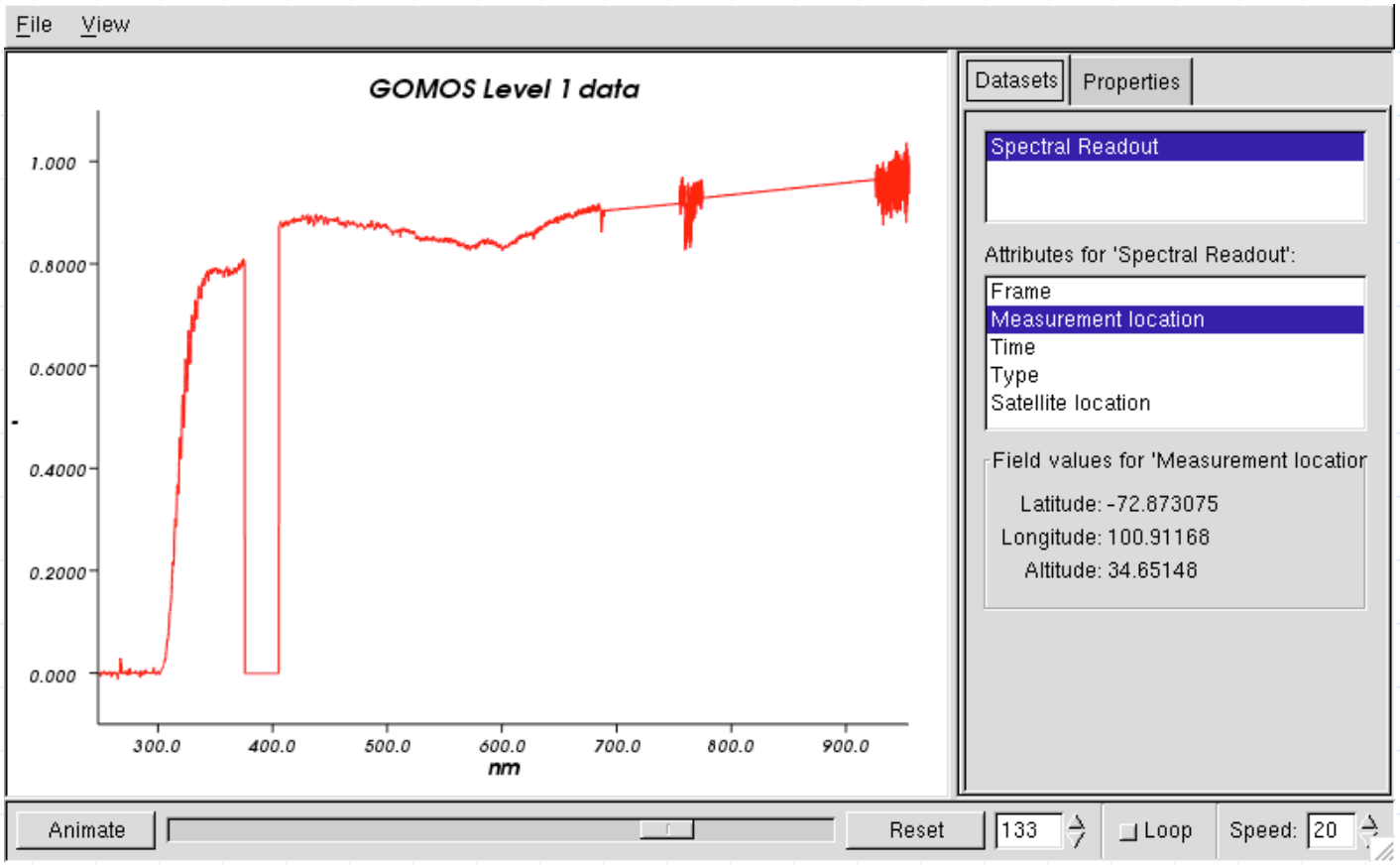








VISAN - 2D plot



The future

Support for more instruments coming:

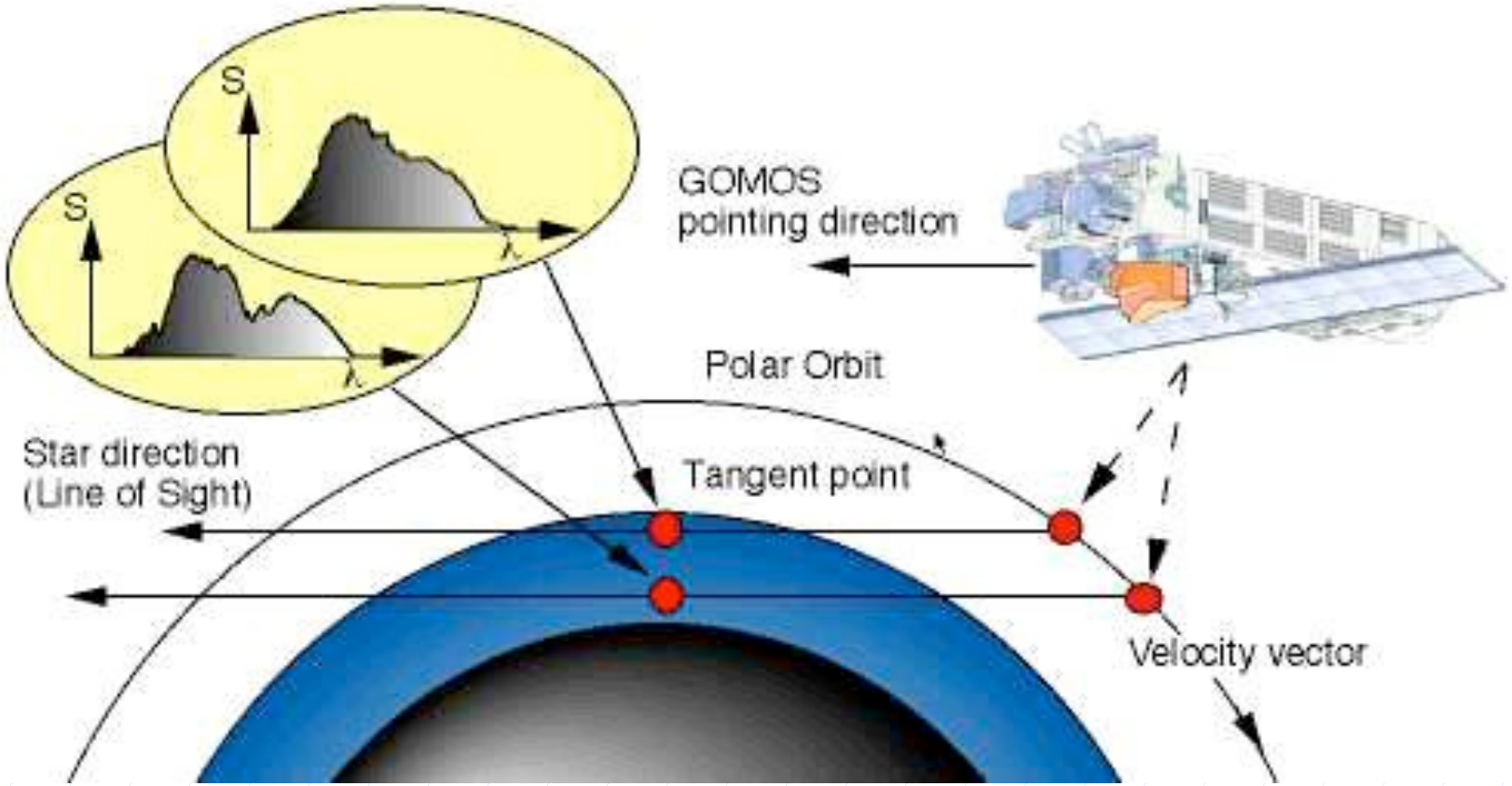
- OMI
- GOME-2
- …

For information and download visit:

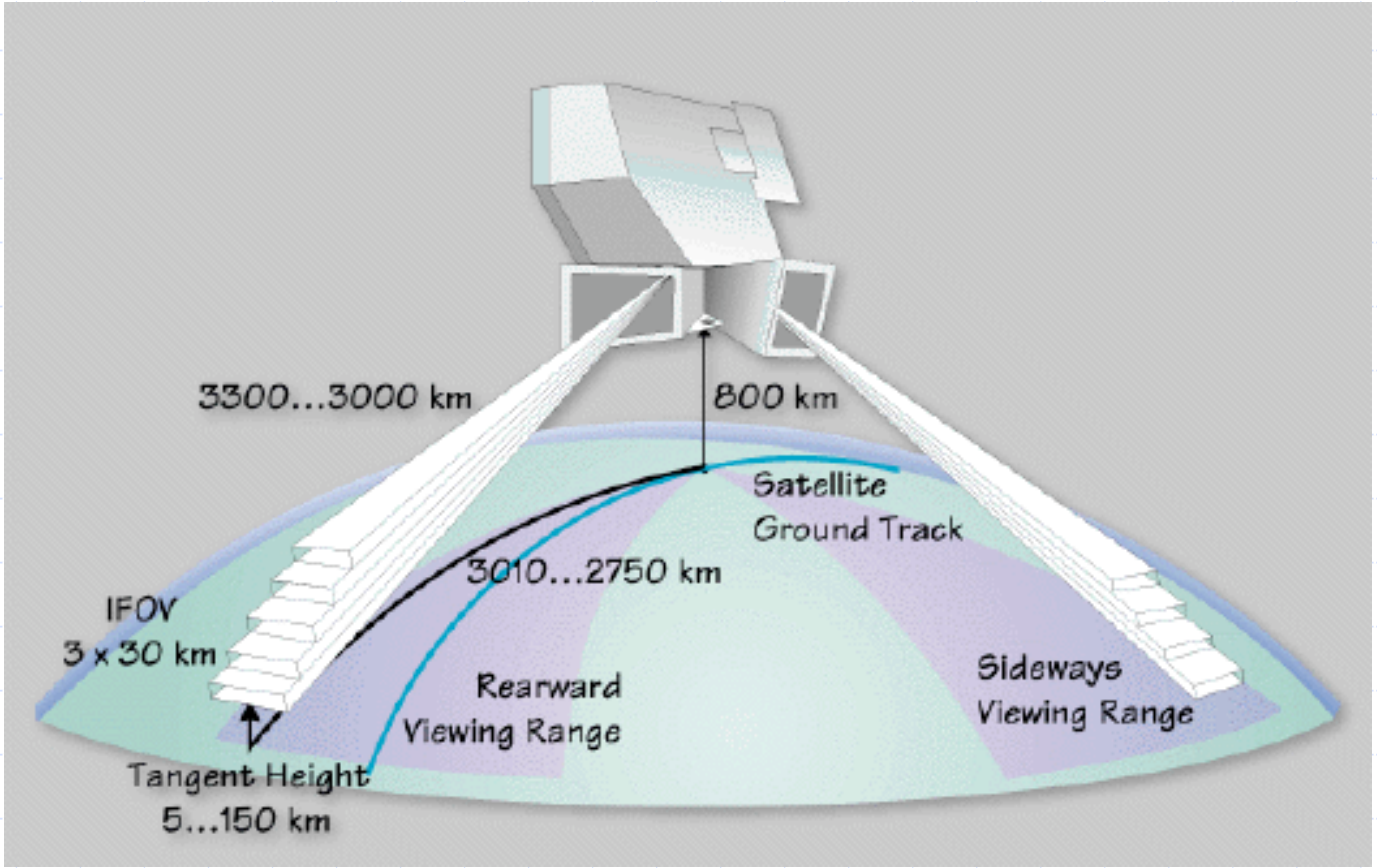
http://www.science-and-technology.nl/beat/

Questions…

GOMOS


S
λ
S
λ
GOMOS
pointing direction
Polar Orbit
Star direction
(Line of Sight)
Tangent point
Velocity vector

MIPAS


3300...3000 km
800 km
Satellite
Ground Track
3010...2750 km
IFOV
3 x 30 km
Rearward
Viewing Range
Sideways
Viewing Range
Tangent Height
5...150 km

SCIAMACHY