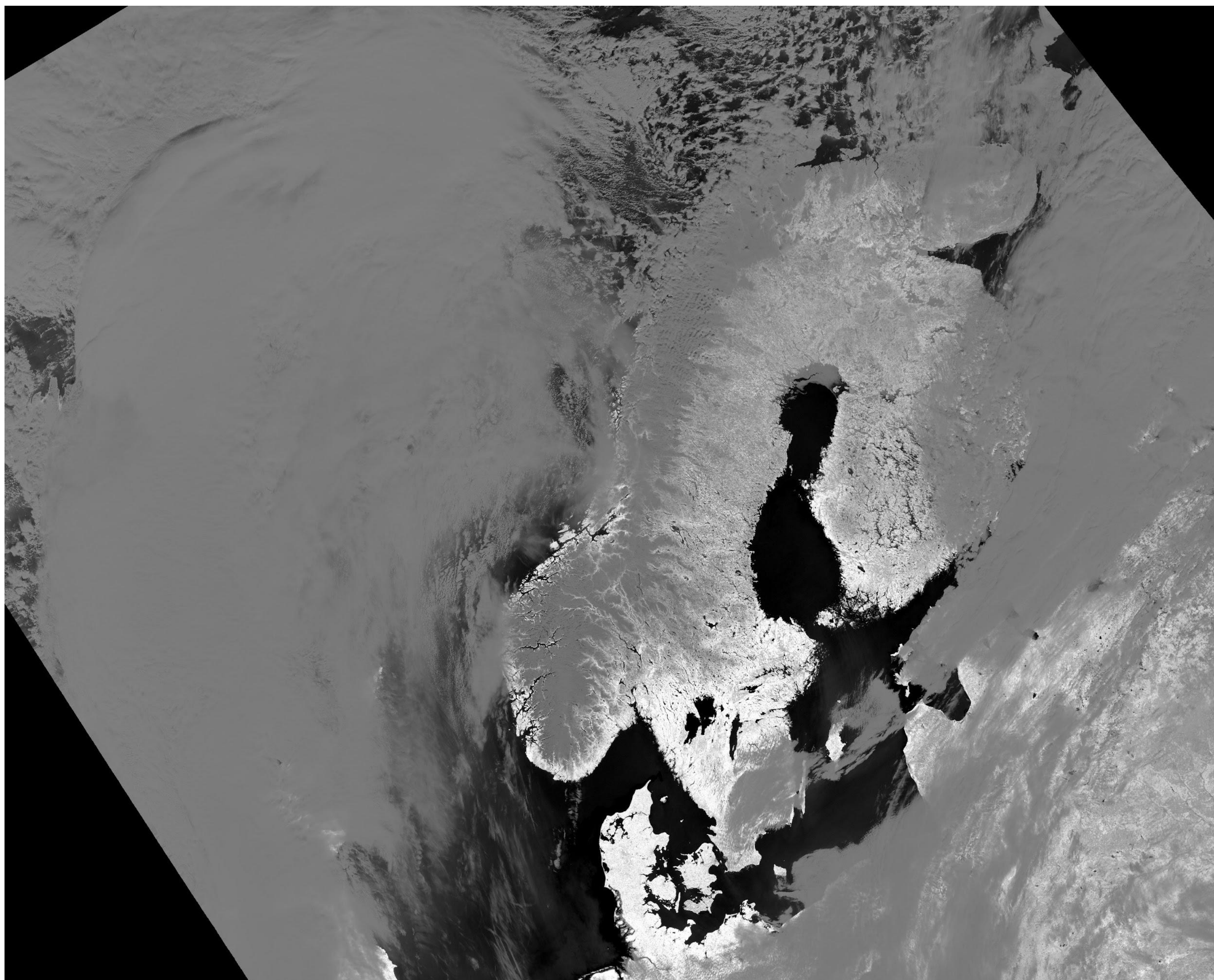


A Python Library for Weather Satellite Imagery

David Hoese (SSEC), Martin Raspaud (SMHI)

david.hoese@ssec.wisc.edu, martin.raspaud@smhi.se

The PyTroll community has from the beginning strived to make working with weather satellite data easier. Taking over the venerable MPOP, SatPy makes it possible to read data from various satellite file formats, reproject that data, and write scaled data to image and data file formats in a pythonic and intuitive way. SatPy is a new python library designed to make access to this functionality even easier than before. The PyTroll community developing SatPy has existed for more than 5 years and consists of software developers and scientists from around the world. This community started development on the SatPy library in an effort to combine the ease and functionality of the SSEC/CIMSS Polar2Grid project with the simple interface and user group of PyTroll's MPOP library. SatPy is still a fairly new project, but it is growing fast and integration into Polar2Grid has already started.



```
from datetime import datetime
from satpy.scene import Scene

# Find files by metadata
scn = Scene(platform_name="Suomi-NPP",
            sensor="viirs",
            start_time=datetime(2015, 3, 11, 11, 18),
            end_time=datetime(2015, 3, 11, 11, 26),
            base_dir="/data/satellite/Suomi-NPP/SDR")

# Load variables from data files or create composites
scn.load(["M05", "M07"])

# Access common metadata
print(scn["M05"].info["standard_name"])
print(scn["M05"].info["wavelength_range"])

# Treat datasets like a normal numpy masked array
print(rs_scn["M05"].min(), rs_scn["M05"].max())

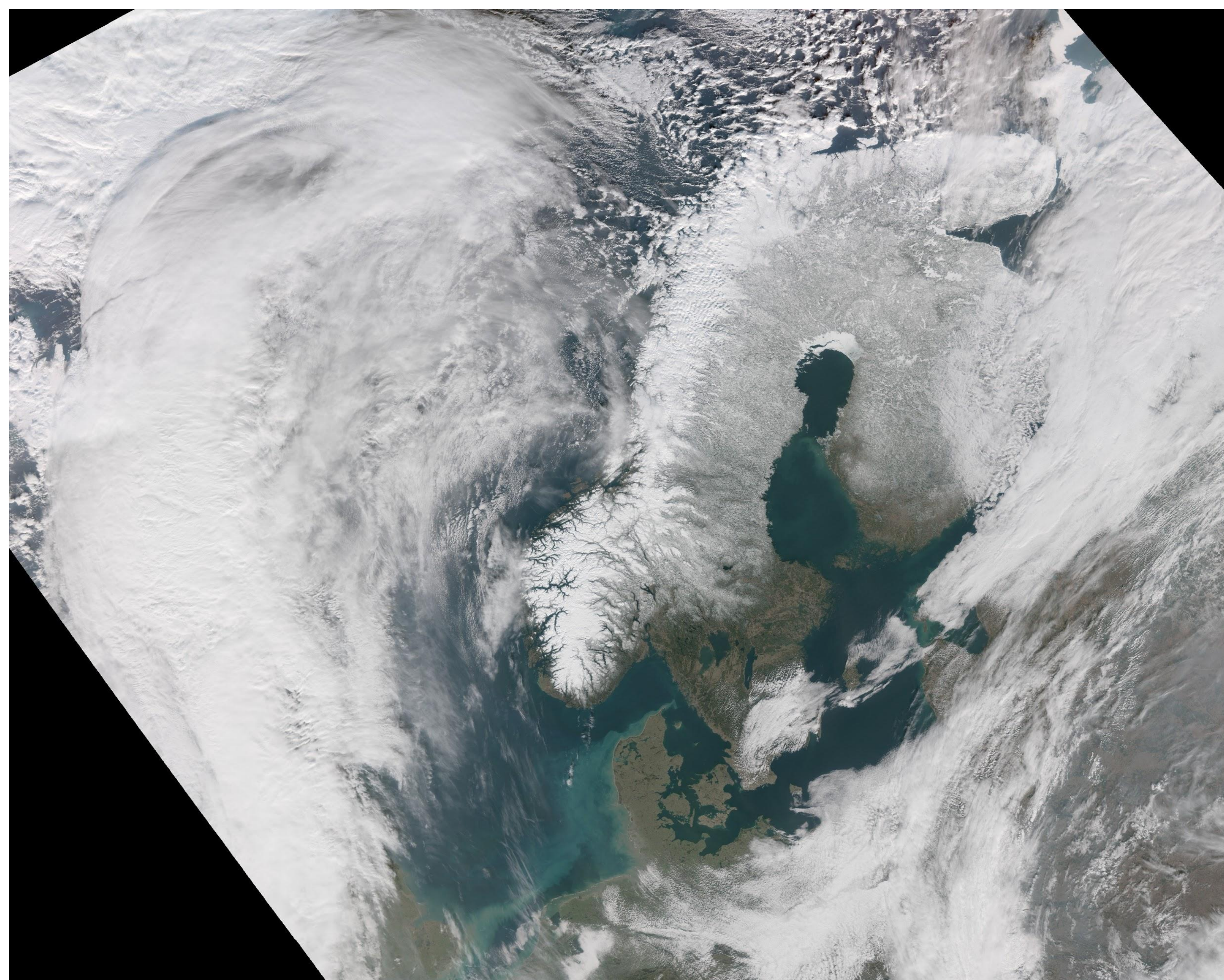
# Resample data to a uniform grid
rs_scn = scn.resample("euron1")

# Create new datasets
rs_scn["ndvi"] = ((newscn["M07"] - newscn["M05"]) /
                  (newscn["M07"] + newscn["M05"]))
# View sample image in a pop up window
rs_scn.show("ndvi")
```

Affiliations

SSEC
1225 W Dayton Street
Madison, Wisconsin
USA

SMHI
Folkborgvägen 17
60176 Norrköping
SWEDEN



```
from glob import glob
from satpy.scene import Scene

# Load data by filenames
files = glob("/data/viirs_pass_1/*.h5")
scn = Scene(reader="viirs_sdr", filenames=files)

# Automatically load composites and their dependencies
scn.load(["true_color"])

# Print dataset information (subclass of numpy MaskedArray)
print(scn["true_color"])
VIIRS/true_color:
  area: swath_2015-03-11T11:16:38.819881_2015-03-11T11:26:35.425125__5376_3200
  end_orbit: 17451
  end_time: 2015-03-11 11:26:35.425125
  platform: NPP
  resolution: 742.0 m
  rows_per_scan: 16
  start_orbit: 17451
  start_time: 2015-03-11 11:16:38.819881
  shape: (5376, 3200)
  ...

# Resample multi-band data to a uniform grid
rs_scn = scn.resample("euron1")

# Save RGB geotiff
rs_scn.save_dataset("true_color")
```

Simple Installation

pip install satpy

GitHub

<https://github.com/pytroll/satpy>

Slack

<https://pytrollslackin.herokuapp.com/>

Supported Output Data

Geotiff
PNG, JPG, GIF, BMP...
Planned: NetCDF/CF
Planned: NinjoTiff, MiTiff

Resampling Algorithms

Nearest neighbour (KD-Tree)
Elliptical Weighted Average
WIP: Gradient search

Supported Input Data

VIIRS SDR
VIIRS L1B
Metop AVHRR L1B
L/HRIT (MFG, MSG)
WIP
MTG FCI, Metop NG
AAPP1B
FY3x
GAC