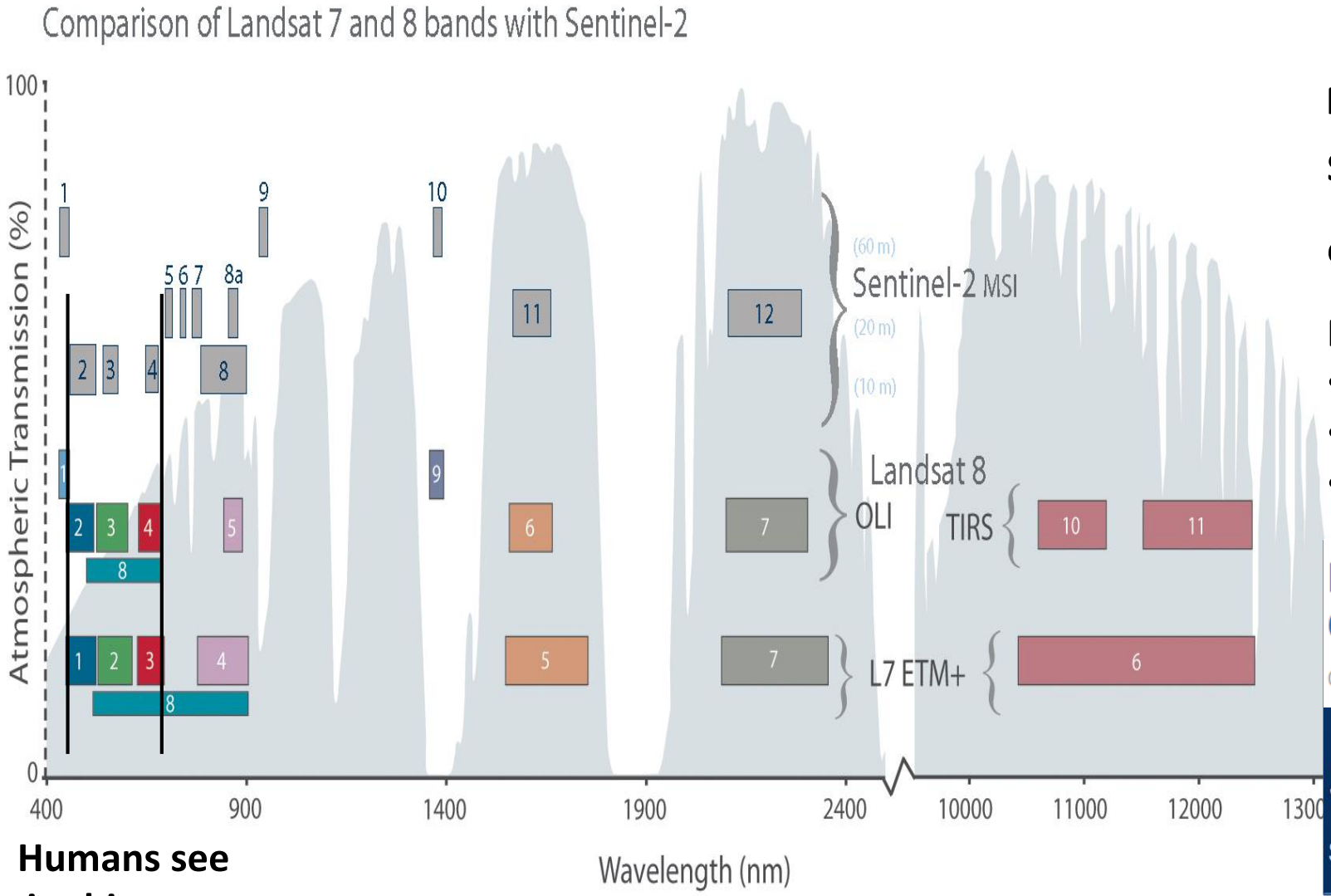
N-D Labeled Arrays and Datasets in Python

Yaron Michael

yaron1205@gmail.com

Satellite photos and Remote sensing



Humans see in this range

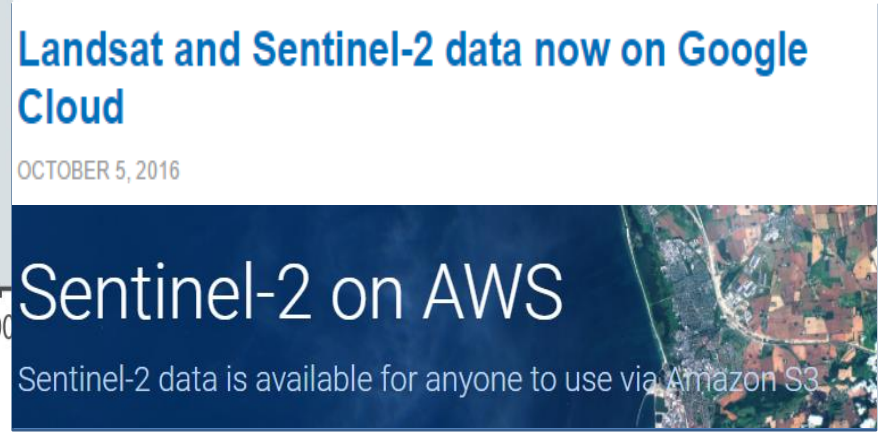
Most cameras and people see in just RGB

Satellites see in lots of "colors"

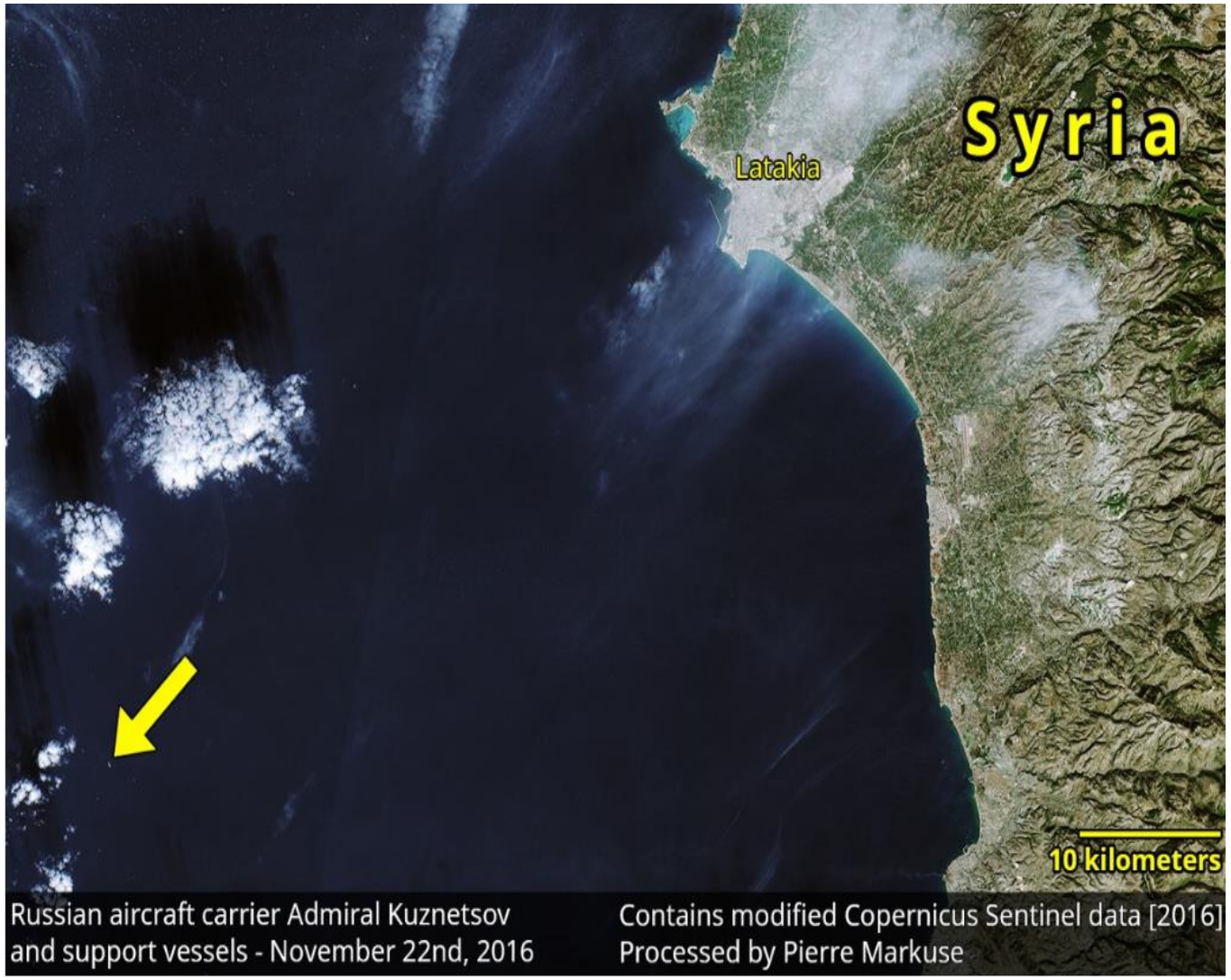
color is channel a band

More bands = more info in the image

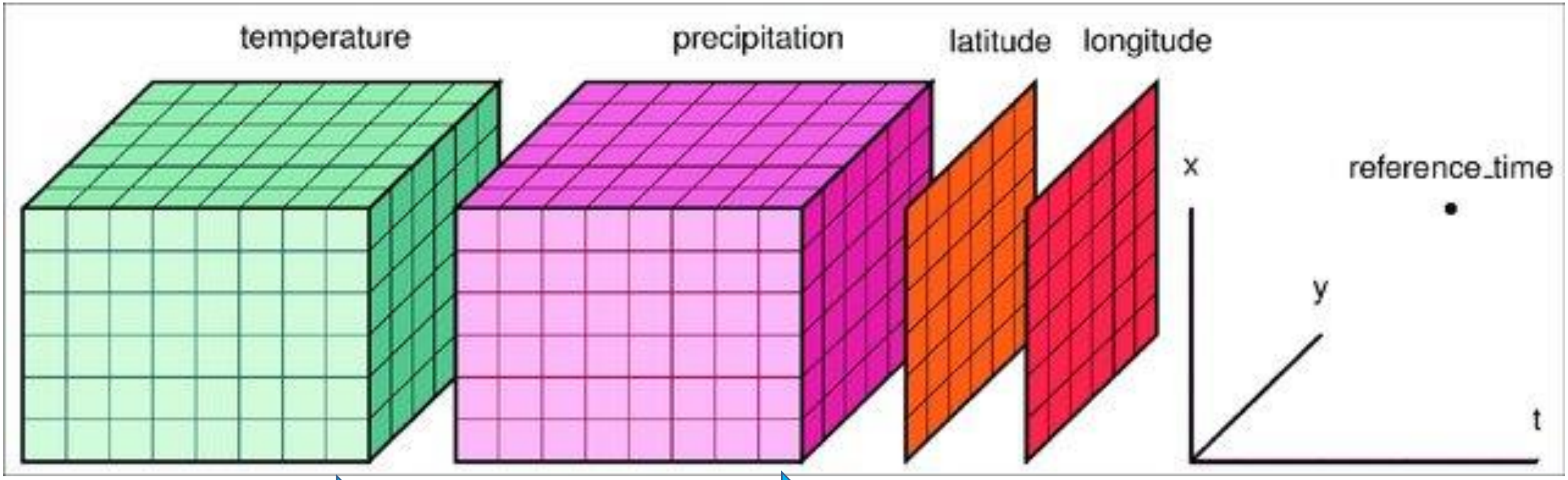
- **See through smoke**
- **See plant Health**
- **And more**



sentinel 2 RGB (10m)



xarray and NetCDF

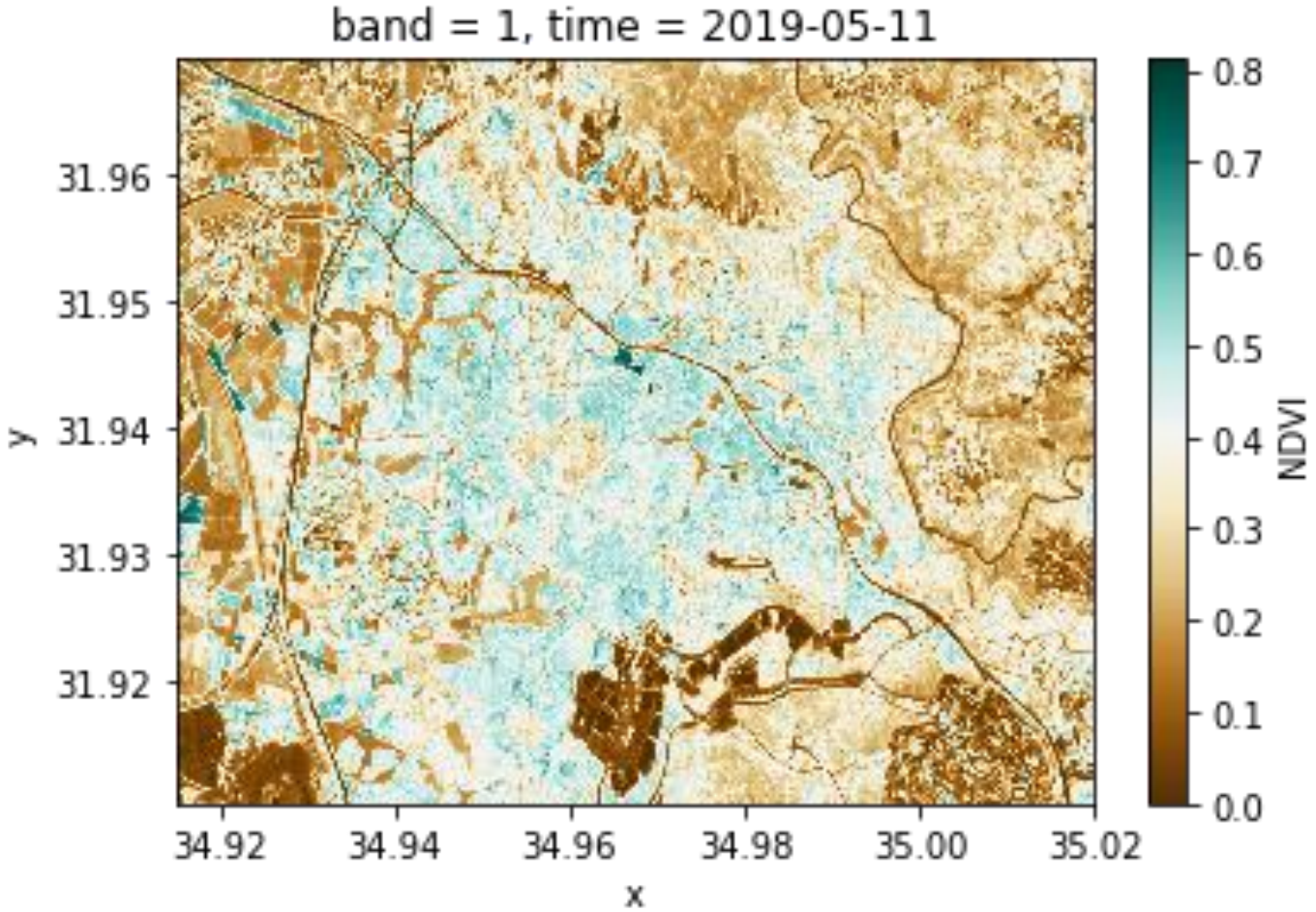


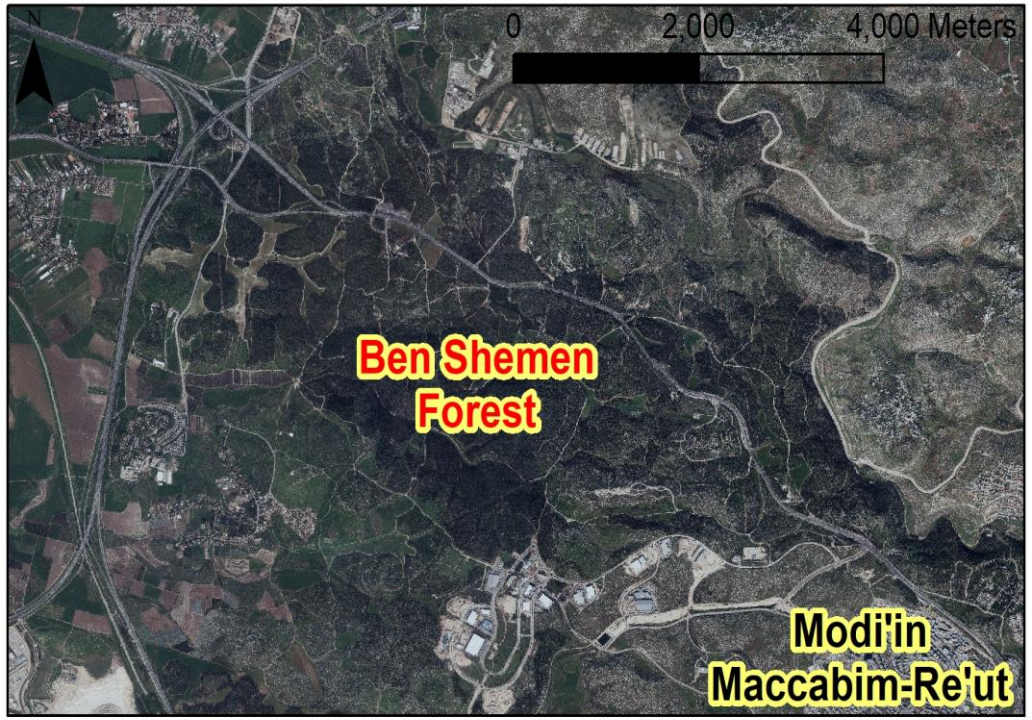
can open and view NetCDF format

ARCGIS use NetCDF format in Space Time Pattern Mining toolbox

xarray and NetCDF

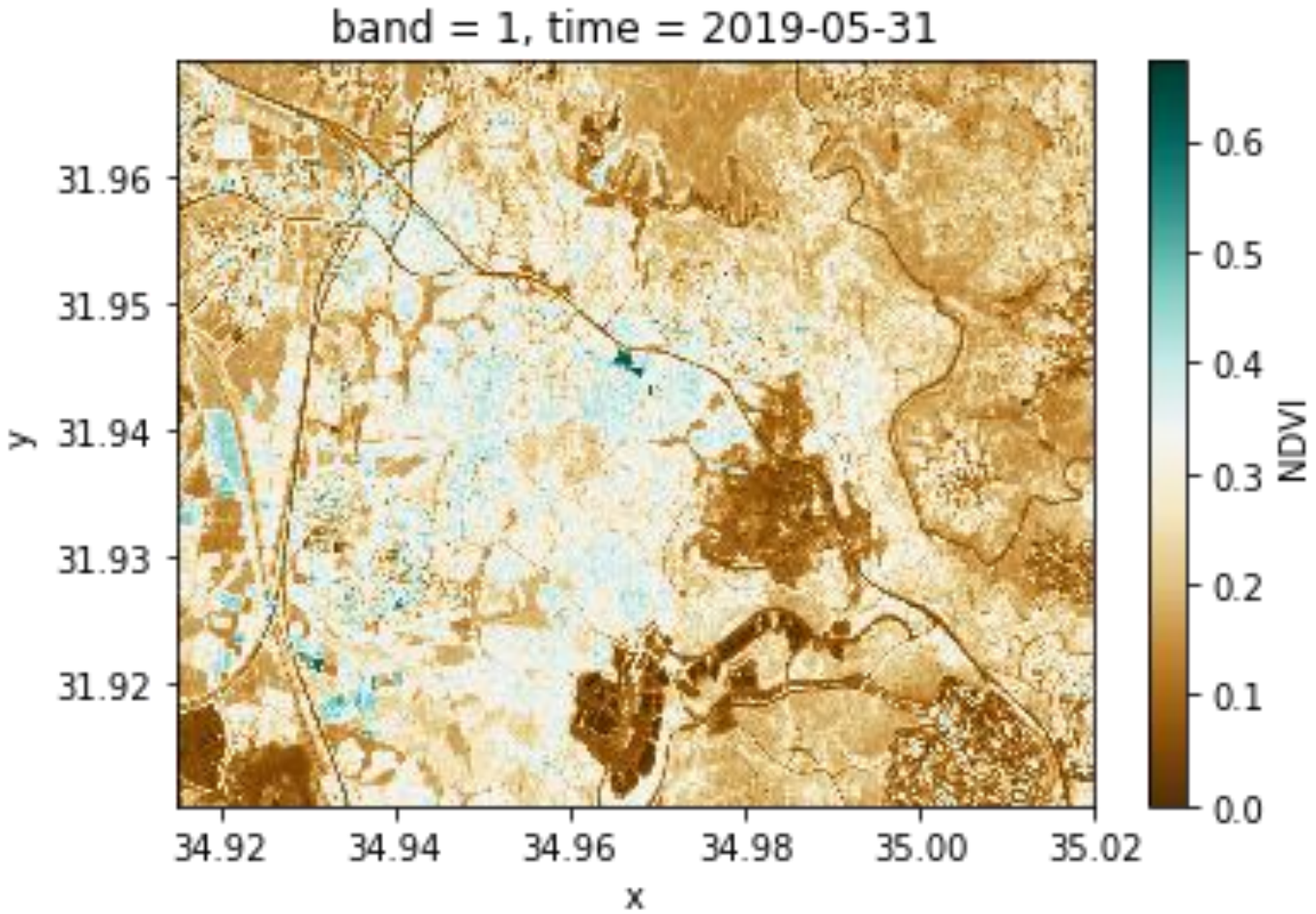
```
da = xr.open_dataset('Sentinel 2 NDVI.nc')
data=da['NDVI']
data.sel(time='2019-05-11').plot(cmap="BrBG")
```

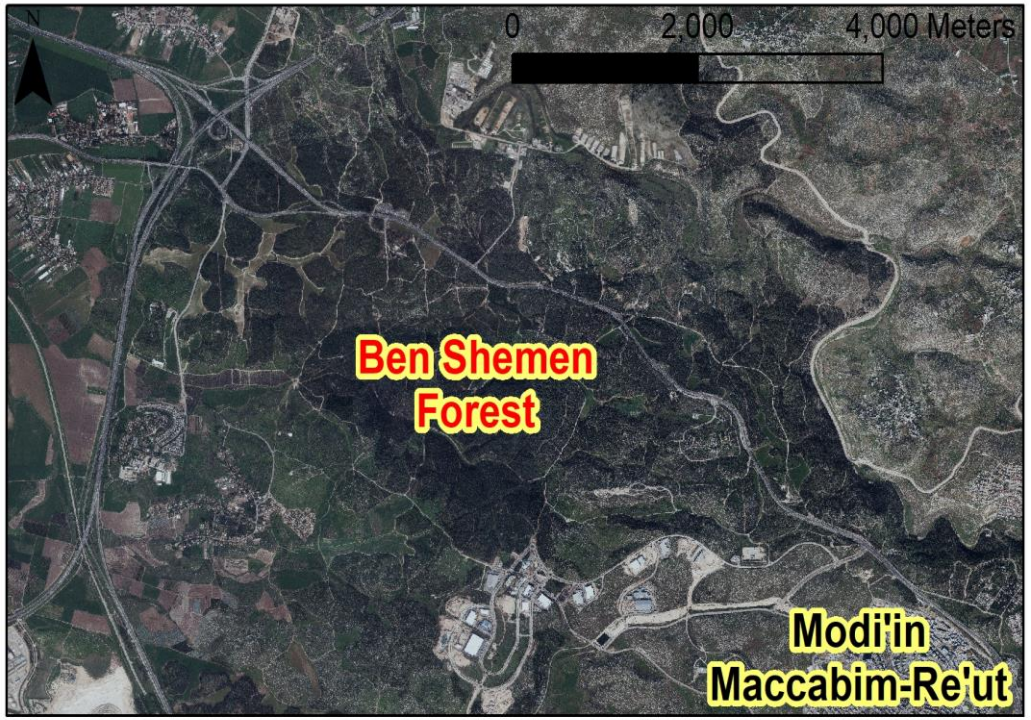




xarray and NetCDF

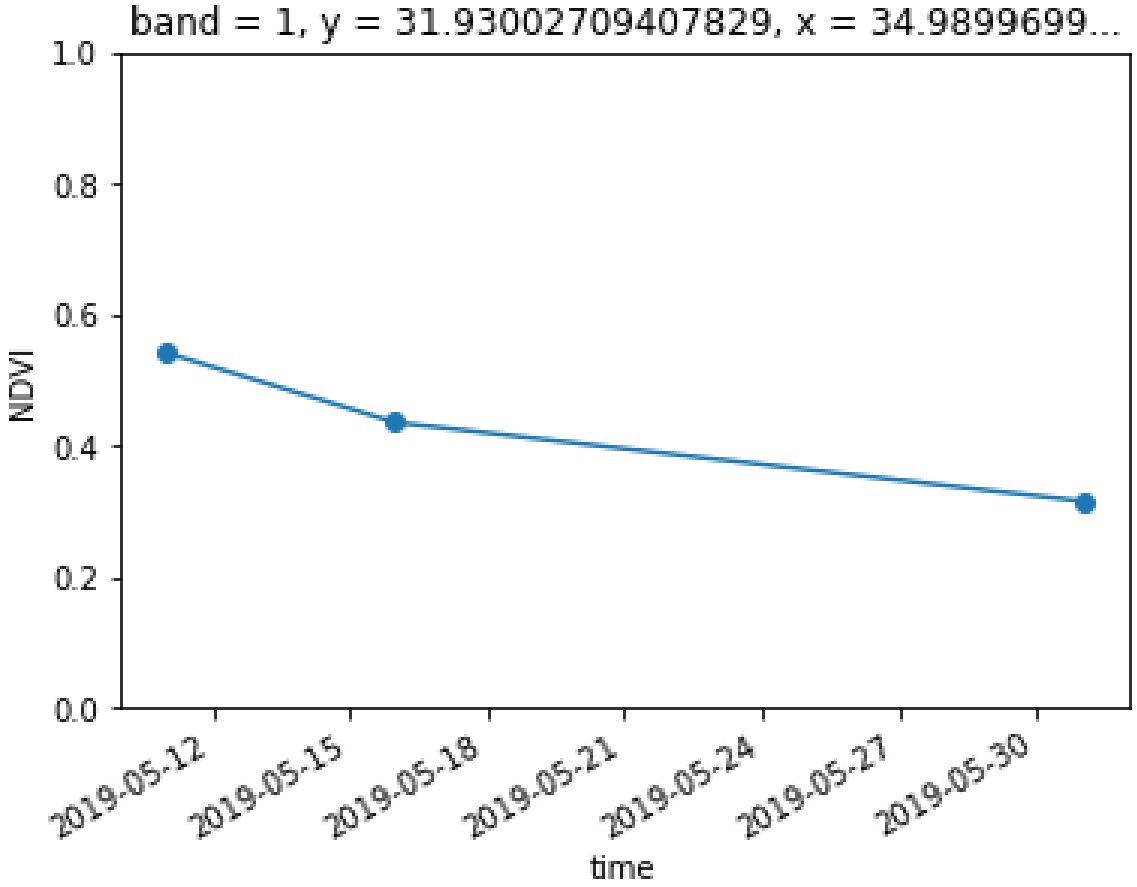
```
1 data.sel(time='2019-05-31').plot(cmap="BrBG")
```

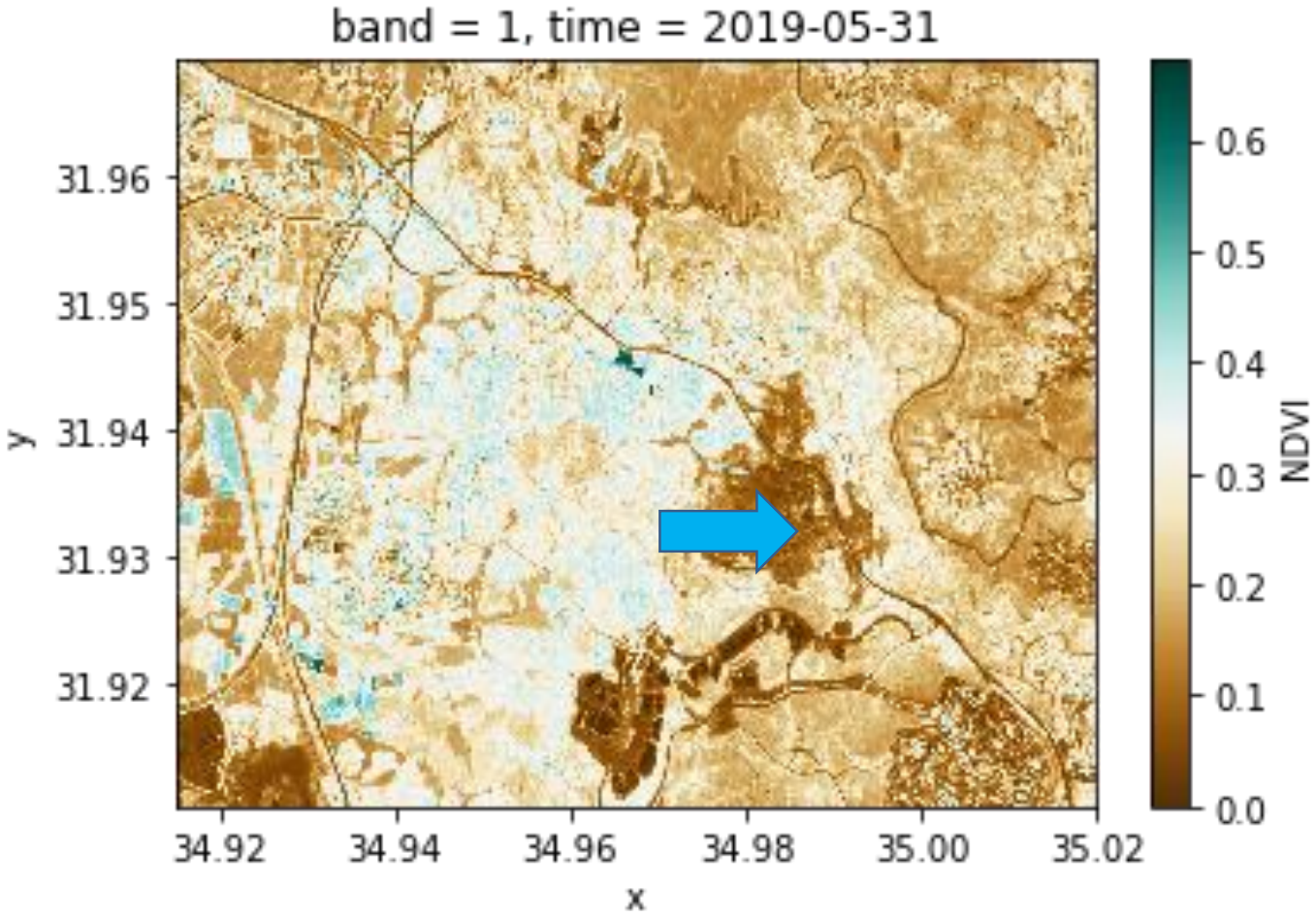




xarray and NetCDF

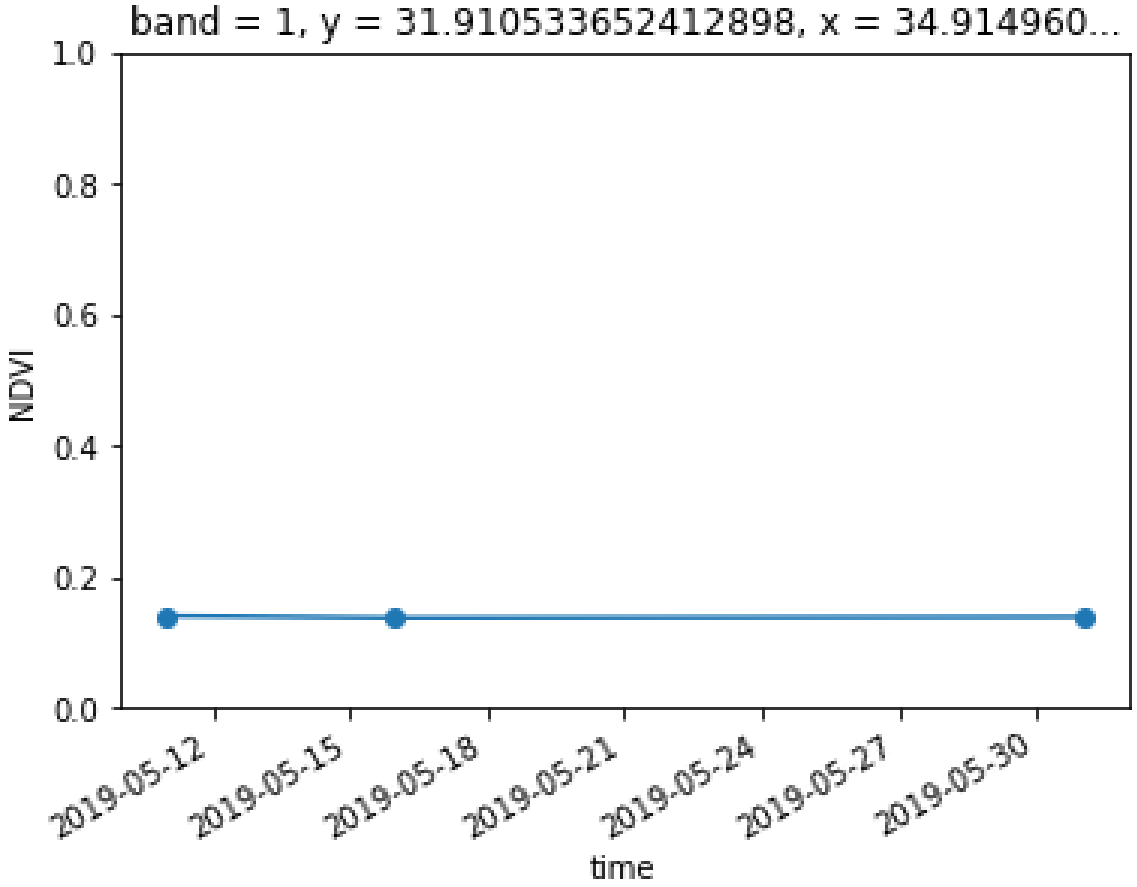
```
1 data.sel(x=34.99, y=31.93, method='nearest').plot( marker='o',ylim= [0,1])
```

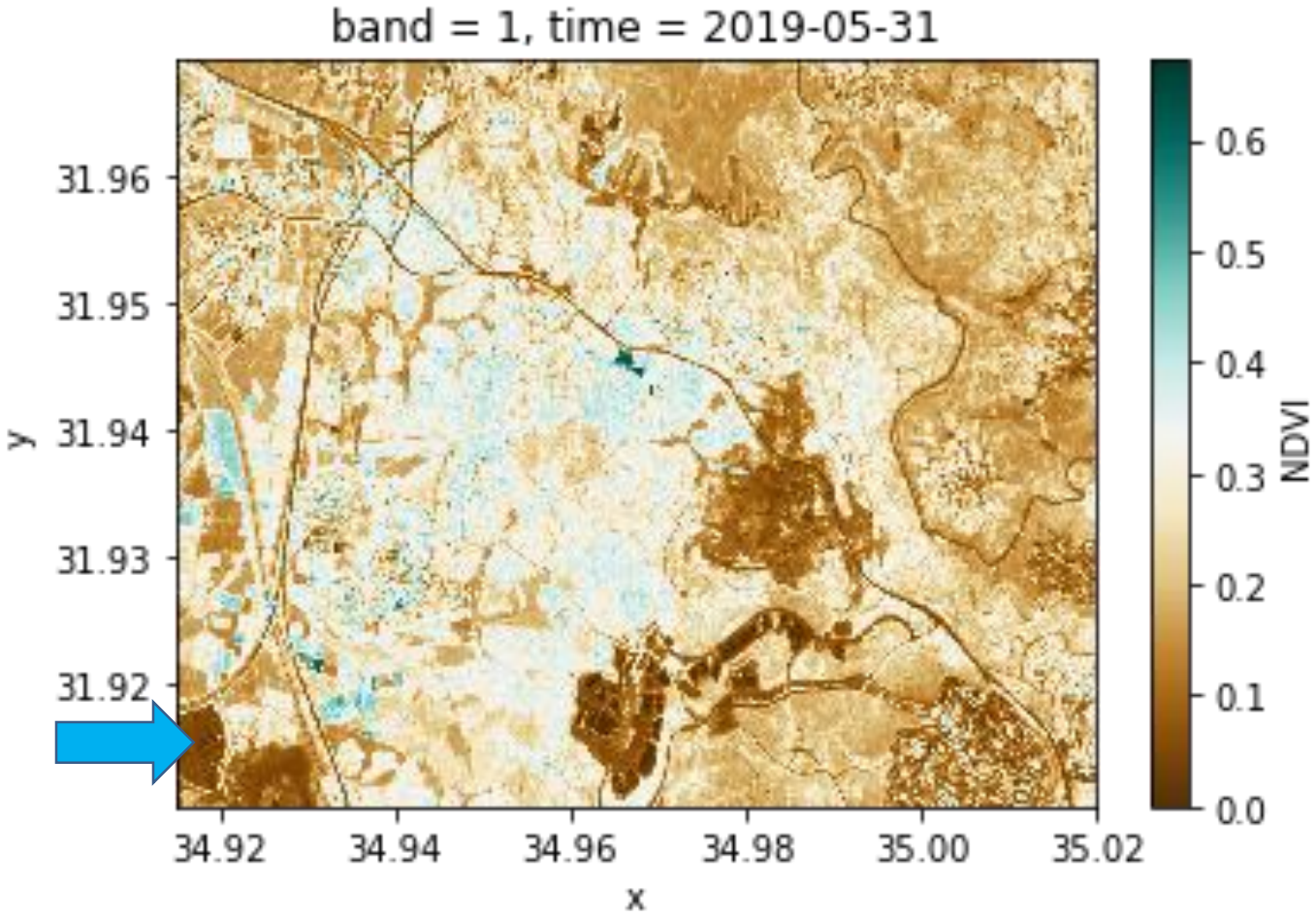




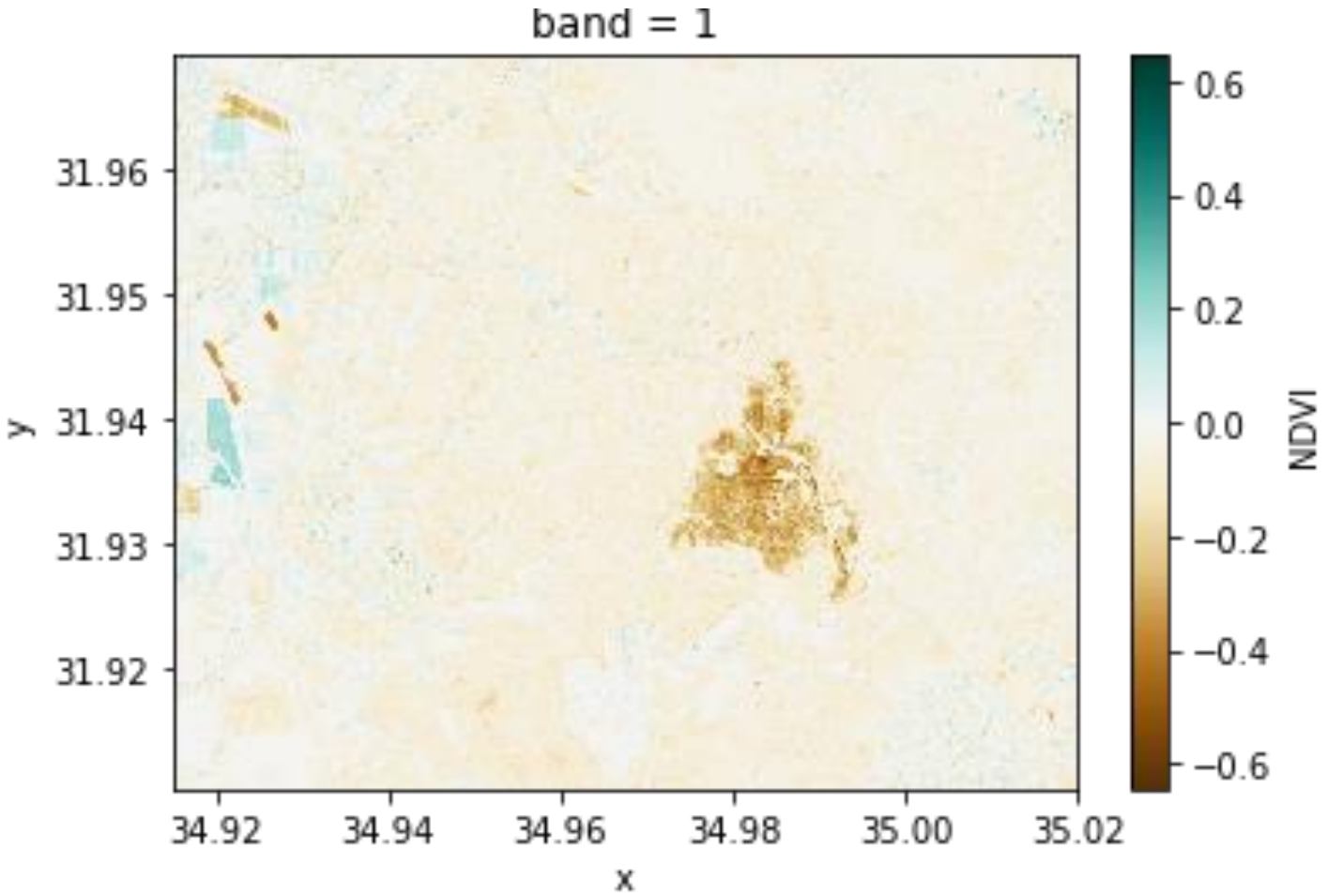
xarray and NetCDF

```
1 data.sel(x=34.91, y=31.01, method='nearest').plot( marker='o',ylim= [0,1])
```





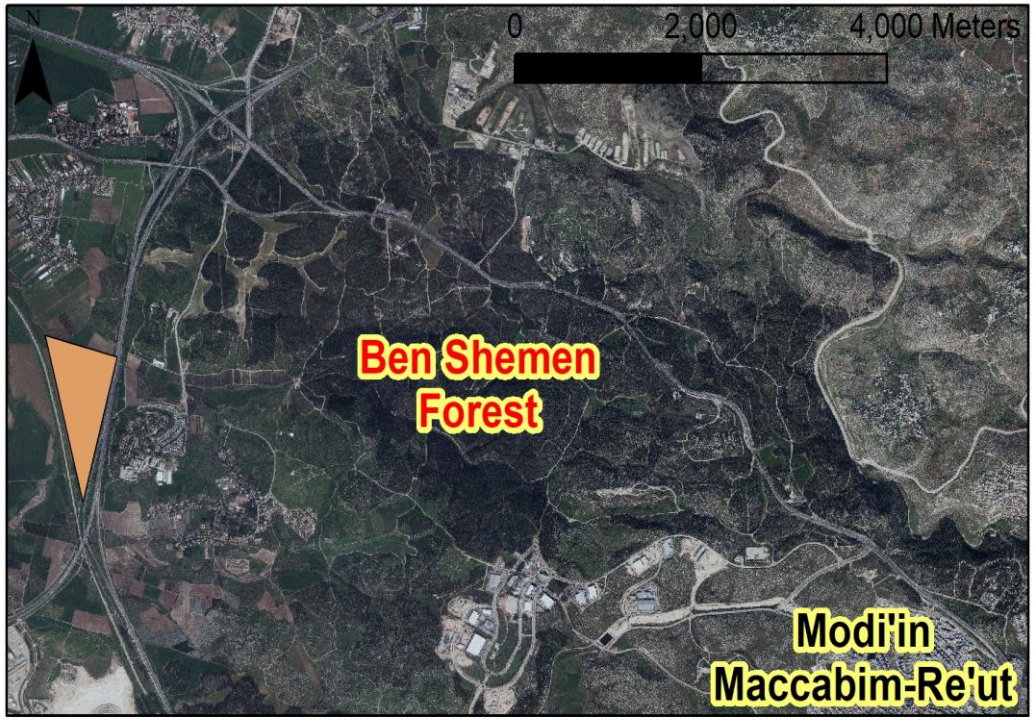
```
After_the_Fire=data.sel(time='2019-05-31')
Before_the_Fire=data.sel(time='2019-05-16')
Where_was_the_fire=  After_the_Fire - Before_the_Fire
Where_was_the_fire.plot(cmap="BrBG")
```



xarray and INCA

```
import xgeo
da=xr.open.dataset()
dsout=da.geo.subset(vector_file = "polygon.shp")
dsout.geo.stats()
```

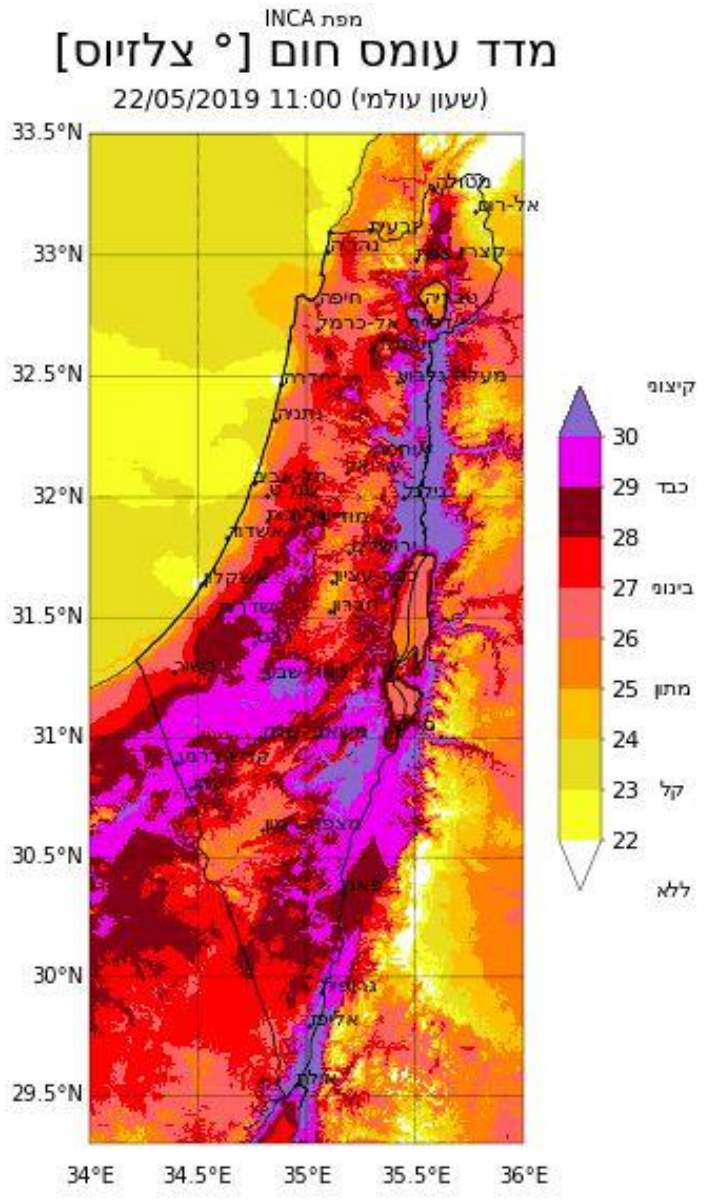
band	time	NDVI_mean	NDVI_std	NDVI_min	NDVI_max
1	2019-05-11	0.355246	0.112137	0.084895	0.759119
	2019-05-16	0.277801	0.070474	0.097070	0.601854
	2019-05-31	0.283891	0.068523	0.099077	0.526218



Integrated Nowcasting through Comprehensive Analysis (INCA)

- 1km Spatial resolution
- 1 HOUER TIME STEP
- wind speed/Wind direction
- air temperature AND RH Relative humidity

Heat index

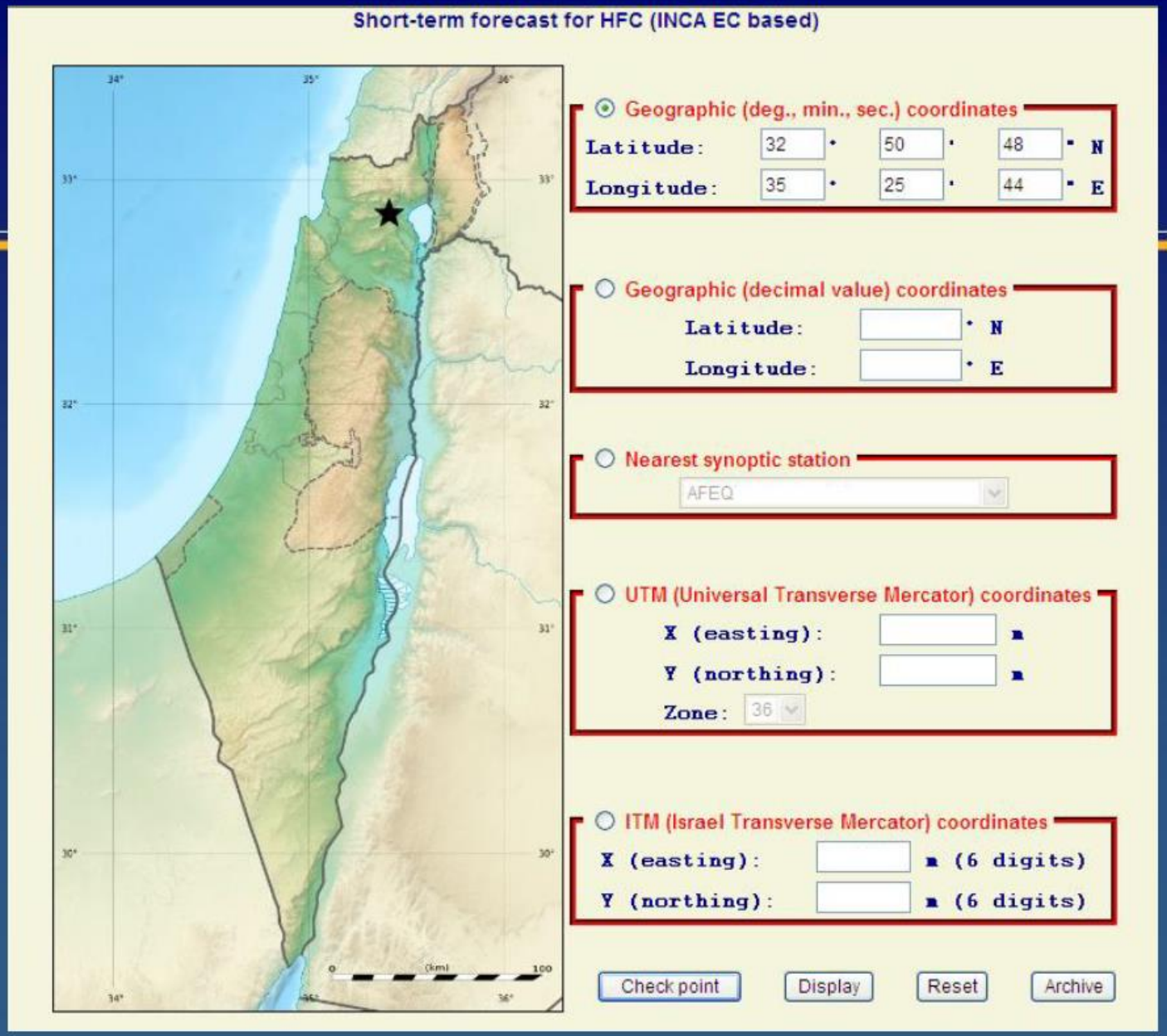


Available for download for the year 2010-2017

IACDC Israeli Atmospheric and Climatic Data Center

INCA

Point forecasting for Chemical Accident or Forest Fire



THE IMS Release the information IN BIN FORMAT WITH PYTHON CODE

- **Exporting time series of one point is very slow 3 minute for 310 day**

- **we have time series million point to Export for Environmental epidemiology study**

solution

- **USE xarray to convert THE DATA TO NETCDF AND TO READ it 3 hours for all point**

INCA Data processing

```
INCA_data_list =["INCA_TT_ANA_201001010600.bil",]
for INCA_data in inca_file:

    yyyy = int(INCA_data[12:16])
    mm = int(INCA_data[16:18]) # get month
    dd = int(INCA_data[18:20]) # get day
    HH = int(INCA_data[20:22]) # get hour
    MM = int(INCA_data[22:24]) # get minute
    date = datetime(yyyy,mm,dd,HH,MM,0)

    Array = readbil(INCA_data,100.)
    INCA = xr.DataArray(INCA_Array, dims=['x', 'y'], coords={'time':date})
    INCA_data_list.append(INCA_Array_with_date)

INCA_REDY_xr = xr.concat(INCA_data_list, 'time')
INCA_REDY_xr.to_netcdf('INCA.NC')
```

More file

Get the date of the Raster

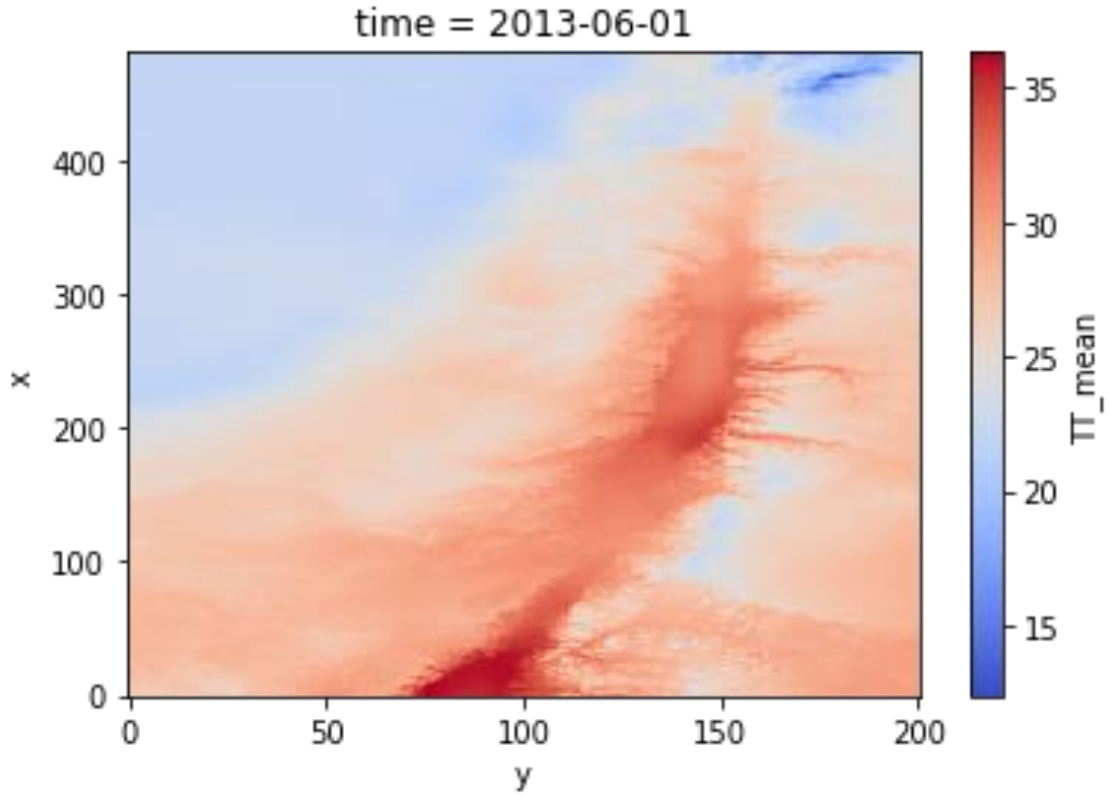
xarray and INCA

```
1 INCA = xr.open_mfdataset('INCA.nc')
2 DATA = INCA["TT_mean"].sel(time=slice('2013-05-31','2014-05-31'))
3 DATA.sel(time='2013-06-01').plot(cmap="coolwarm")
```

The IMS give the XY coordinate in wgs84 for each point in the raster
So GDAL :)

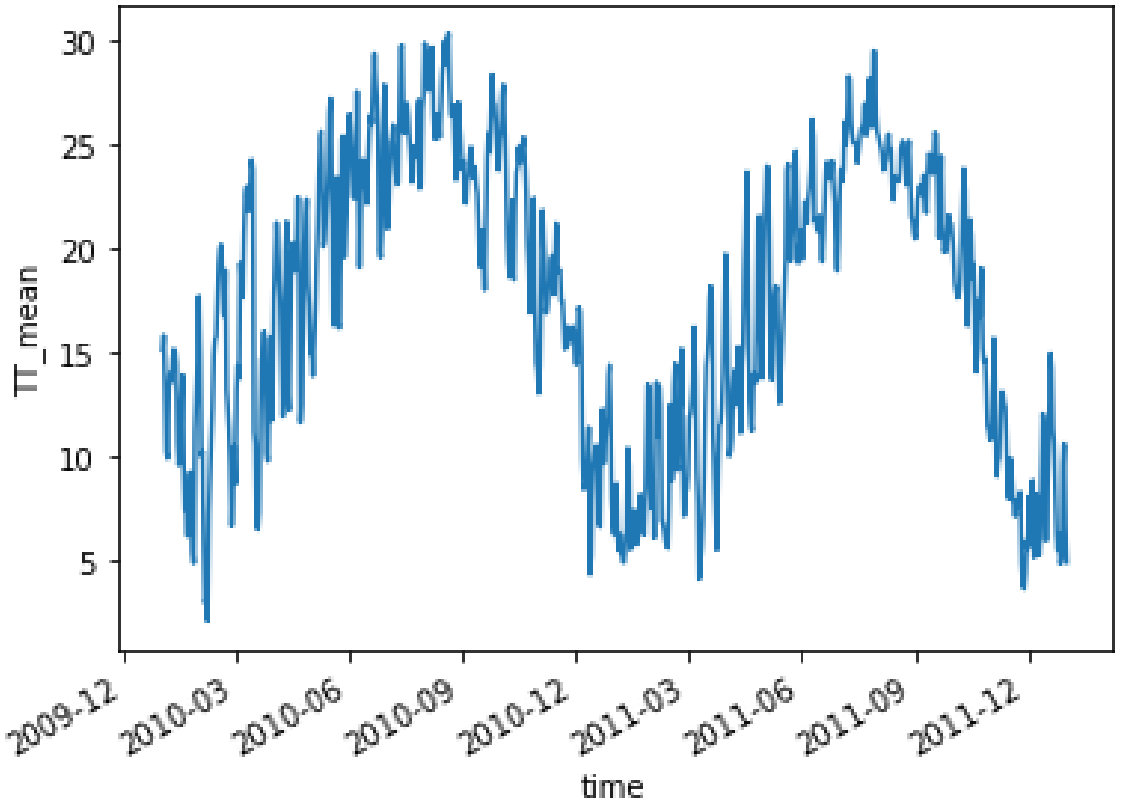
↓

INCA GEOTIF



xarray and INCA

```
1 INCA = xr.open_mfdataset('INCA.nc')
2 Time_series = INCA["TT_mean"].sel(time=slice('2010-01-01','2012-01-01'))
3 Time_series.sel(x=3, y=4).plot()
```



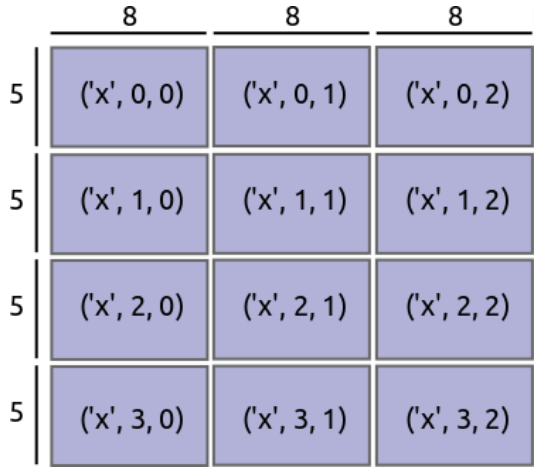
Every operation in xarray is parallelized with Dask

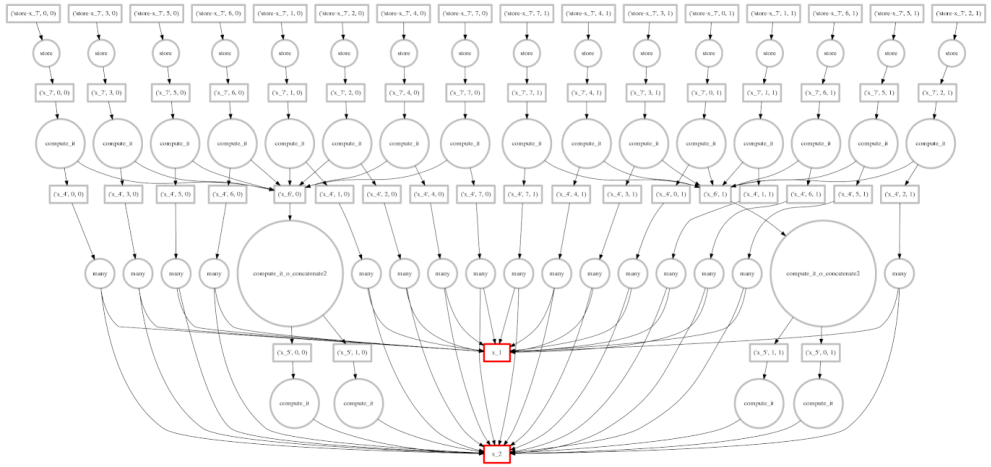
Dask adds two major features to NumPy:

Parallelized: use all your cores
Out-of-core: streaming operations

Dask scales up (to a cluster) *and* down (to a single machine).

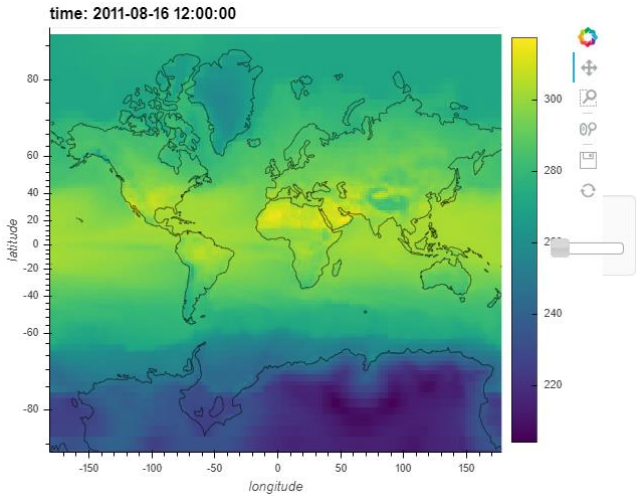
To use Dask in xarray, users specify `chunks` or call `open_mfdataset()`.



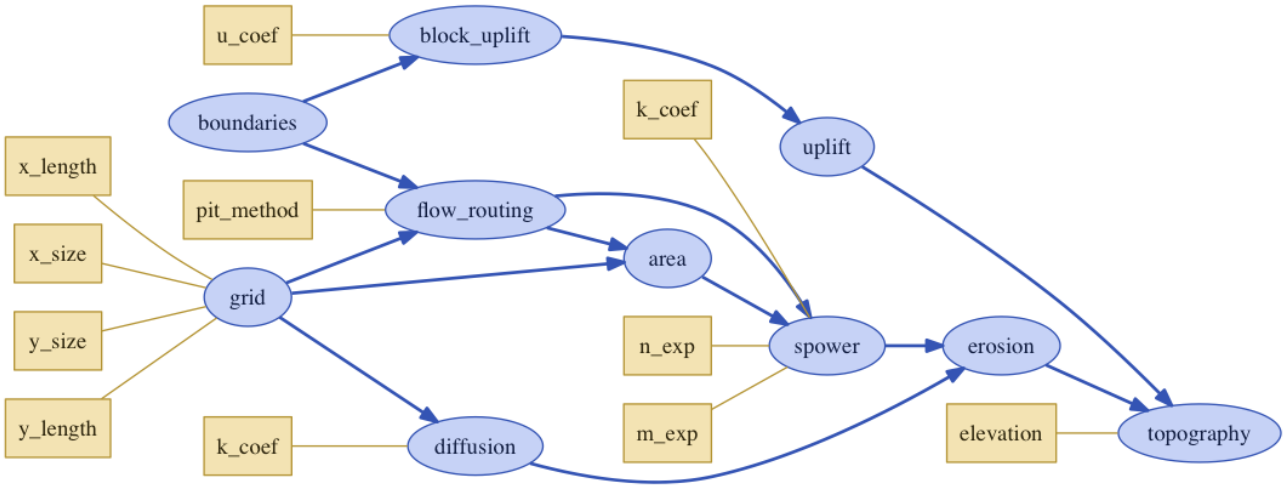


xarray related projects

- Datashader, geoviews, holoviews, : visualization packages for large data



- xarray-simlab: xarray extension for computer model simulation



THE END

yaron1205@gmail.com