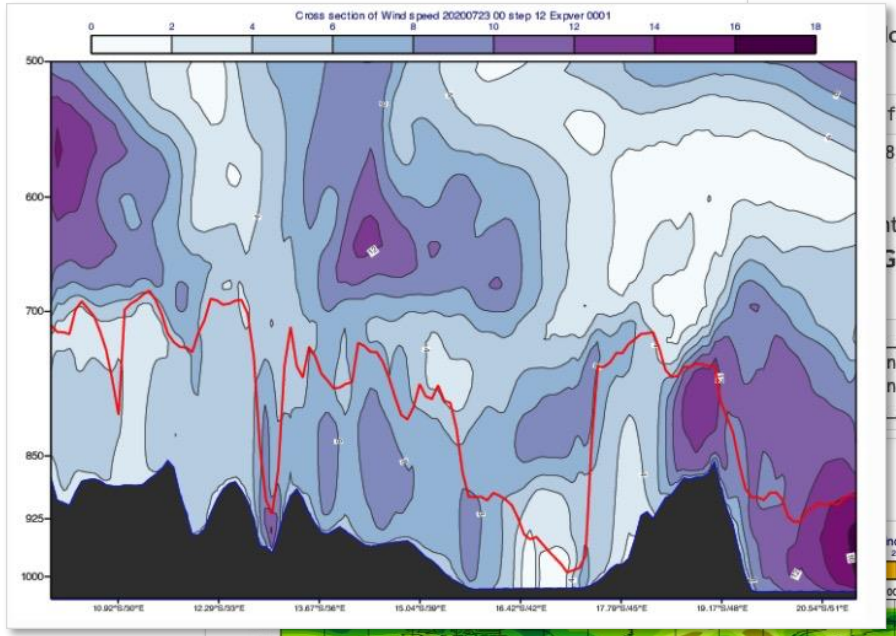
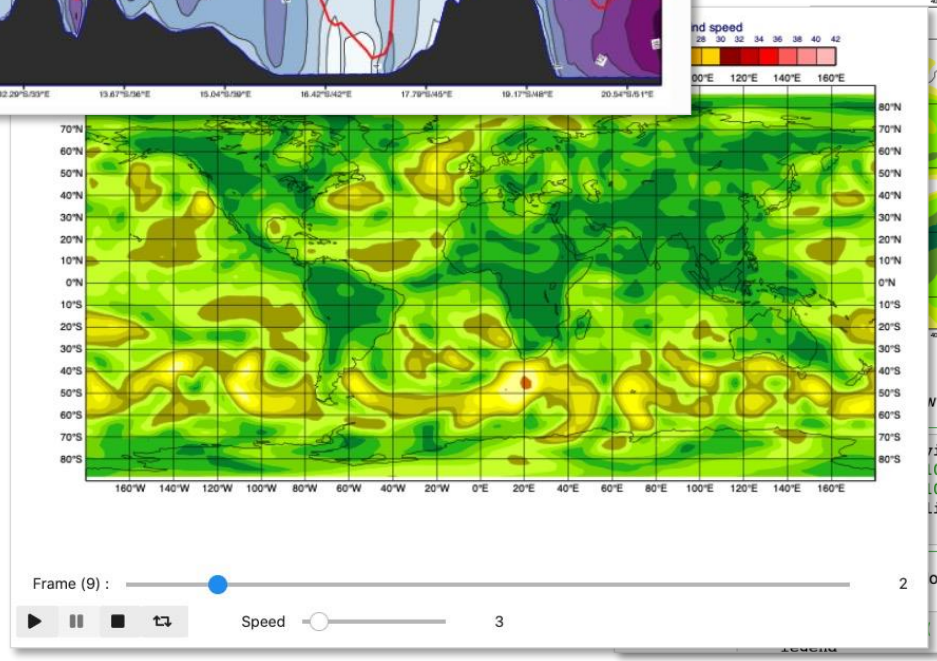
NWP : An Introduction to Metview for Data Analysis in Python

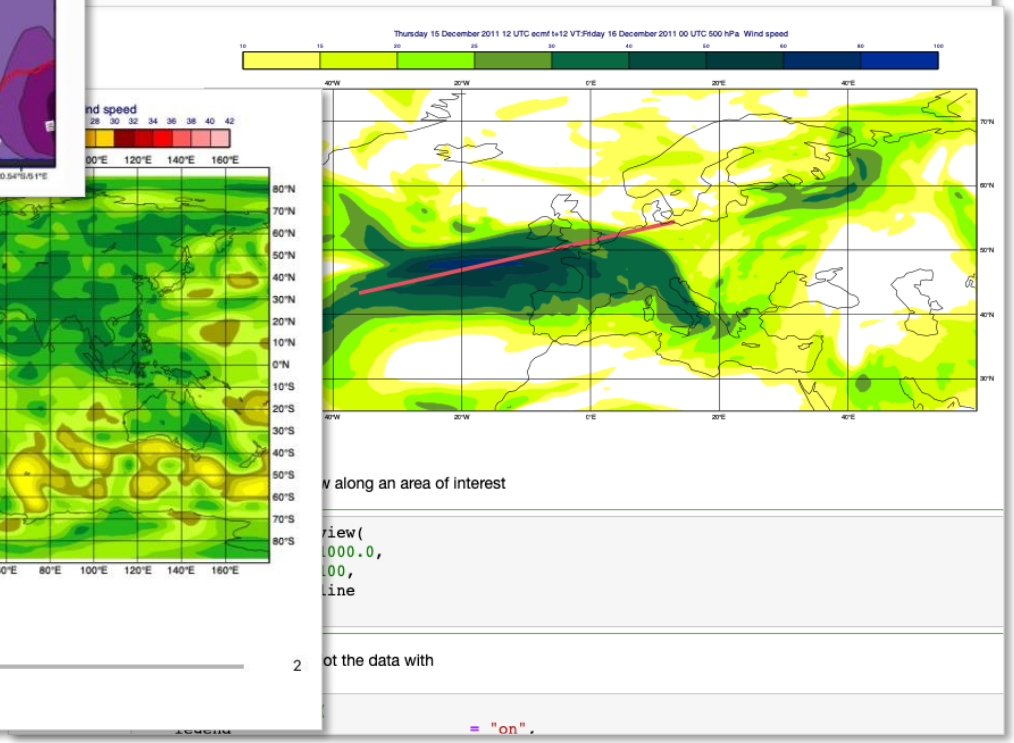
November 14, 2023

Iain Russell
Sándor Kertész

Development Section, ECMWF







Finding a range of extreme values

ocations where the value is within 95% of the maximum by supplying a range of values:

```
find(wg0, [max0*0.95, max0])
```

```
8.5, -5.0], [48.5, -4.5], [47.0, -7.0], [47.0, -3.5], [47.0, -3.0]]
```

t to work with these points in Metview, the easiest way is to use the gfind() function to
Geopoints variable:

```
_points = mv.gfind(wg0, max0, max0*0.05)
nt(len(max_points), 'points')
nt('first point:')
_points[0]
```

v along an area of interest

```
view(
1000.0,
100,
line
```

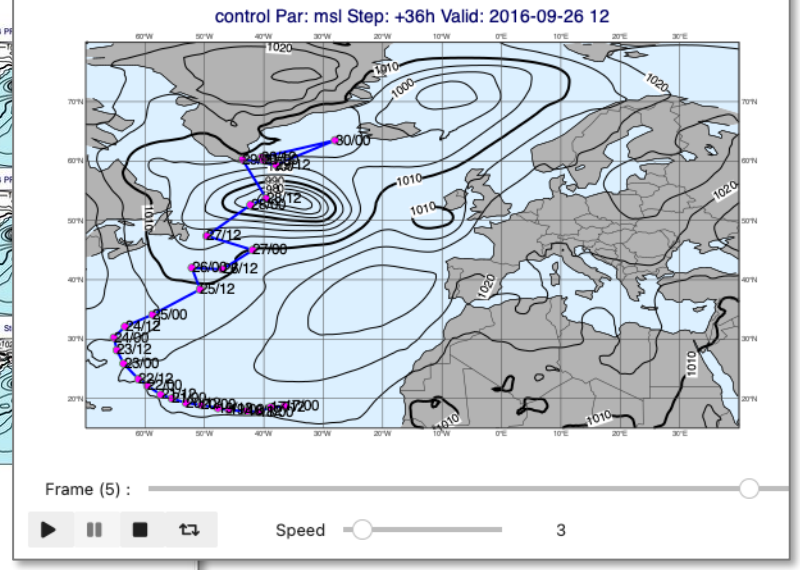
ot the data with

```
= "on",
```

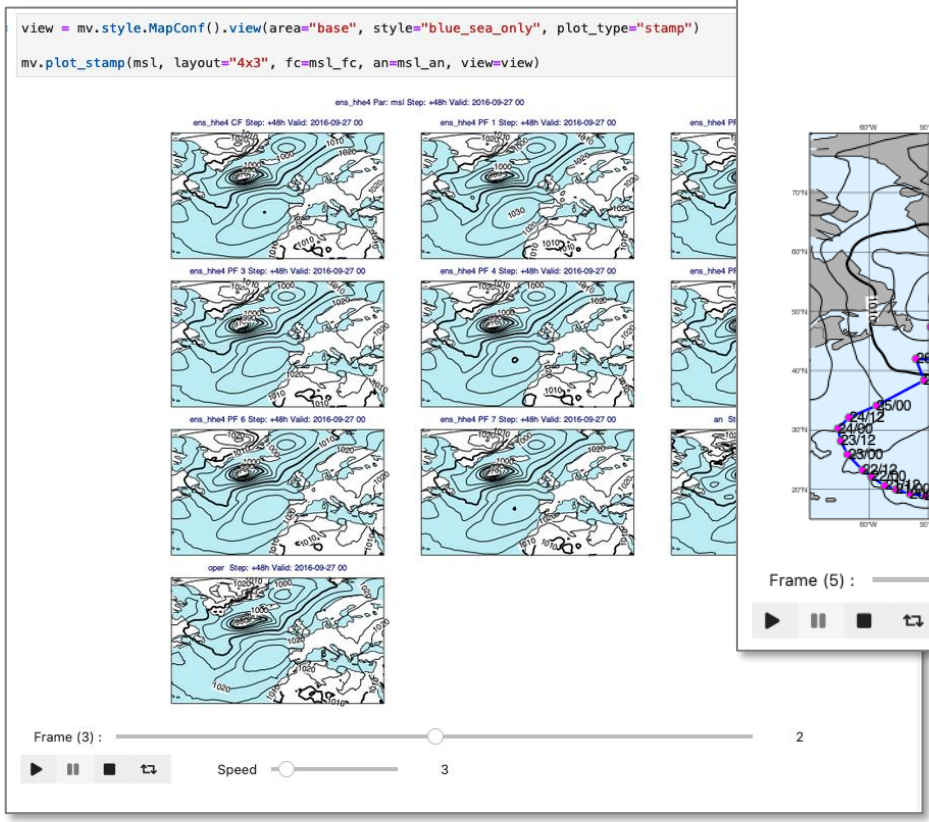
Outline

- What is Metview
- User interface
- Python interface
- How to obtain / install
- Practical exercise using Jupyter notebooks

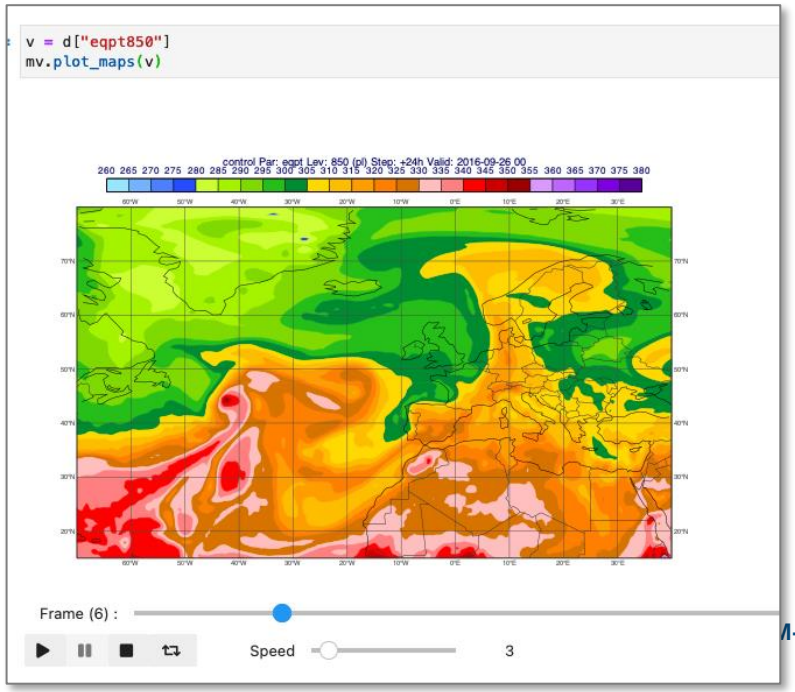
```
import metview as mv

mv.setoutput("jupyter", output_width=600)

ds = mv.load_dataset("oifs_2021")
run = mv.date("2016-09-25 00:00")
step = [0, 12, 24, 36, 48]
d = ds["control"].select(date=run.date(), time=run.time(), step=step)
msl = d["msl"]

tr = ds["track"].select("final_track_op_an_KARL")

vd = tr.style().update(
    {"graph_line_colour": "blue",
     "graph_symbol_colour": "purple",
     "graph_symbol_height": 0.6})

mv.plot_maps(msl, tr, vd, title_font_size=0.6)
```




```
view = mv.style.MapConf().view(area="base", style="blue_sea_only", plot_type="stamp")

mv.plot_stamp(msl, layout="4x3", fc=msl_fc, an=msl_an, view=view)
```

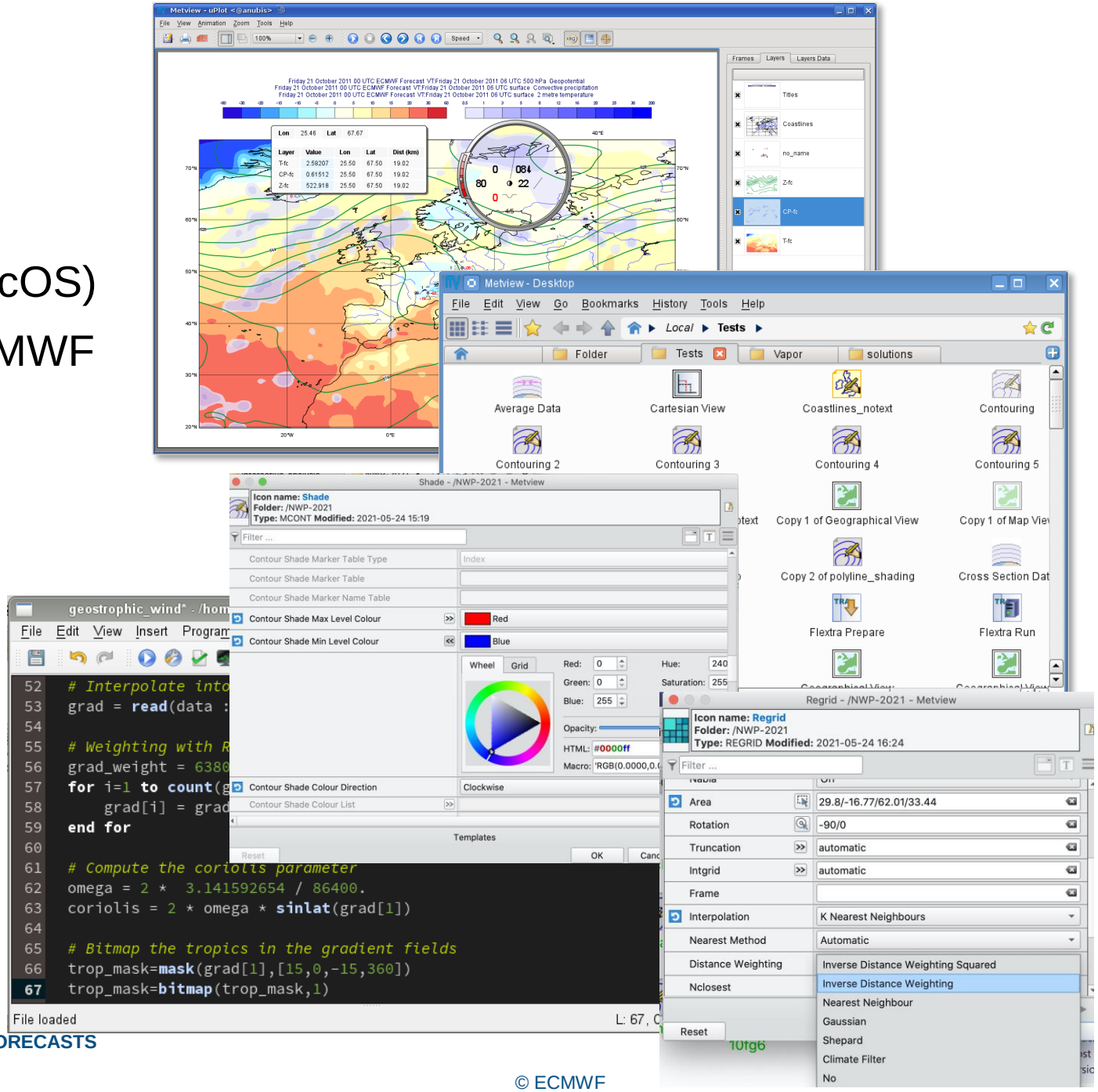



```
v = d["eqpt850"]
mv.plot_maps(v)
```



What is Metview?

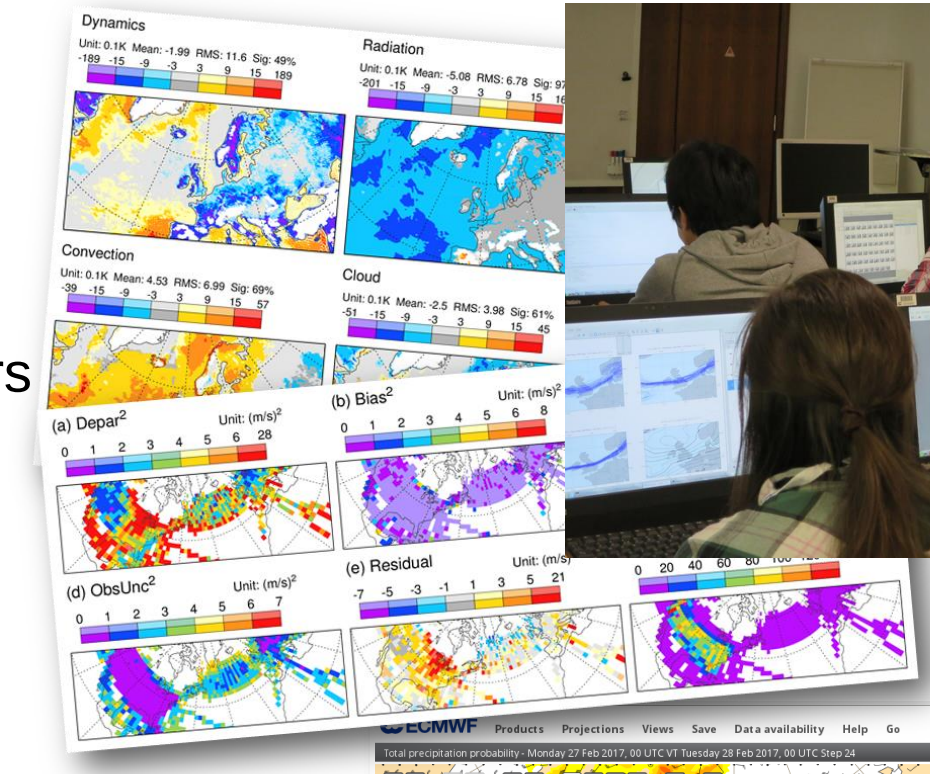
- Workstation software, runs on UNIX, from laptops to supercomputers (including macOS)
- Developed at ECMWF, built on other ECMWF libraries
- Open source, Apache 2.0 license
- Data access
- Data processing
- Data visualisation
- Icon based user interface
- Powerful scripting languages - Macro and **Python**

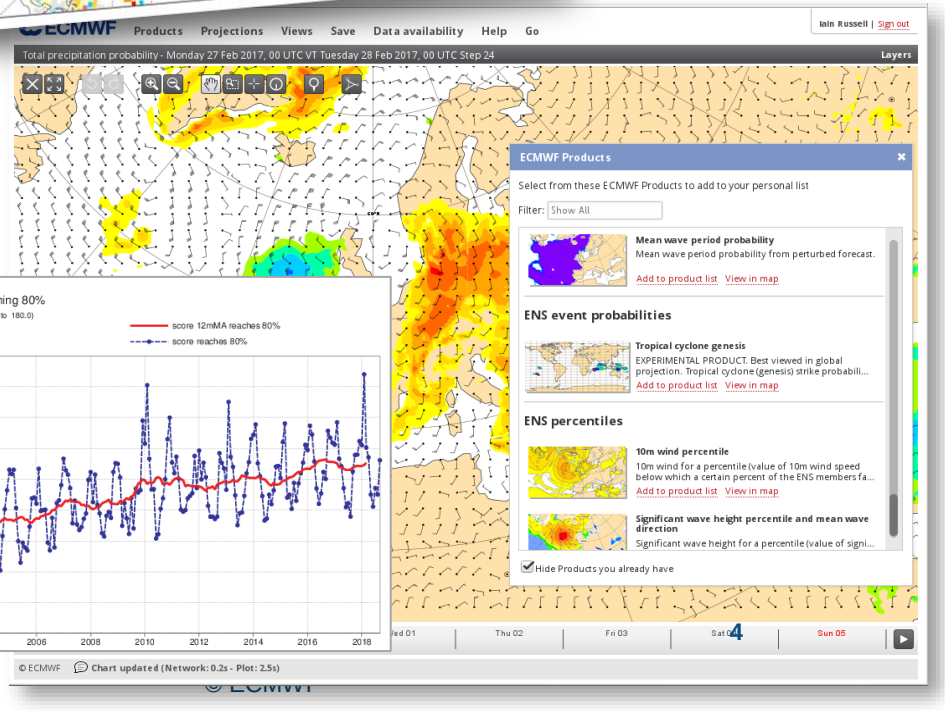


30 years of Metview so far

- Serving users of ECMWF data since 1993

- Used daily by many analysts and researchers
  - inside and outside ECMWF
  - also by commercial users of our data

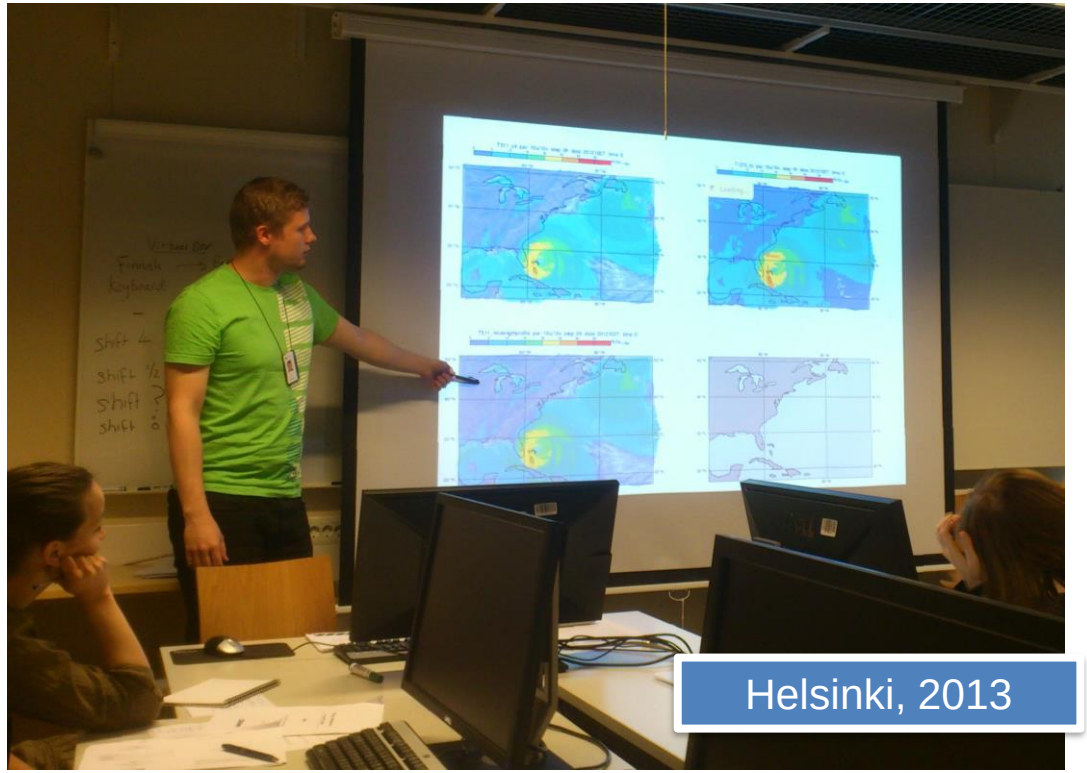




- Some large developments, e.g. the Diagnostics Toolbox, OpenIFS workshops, Quaver (verification package) are based on top of Metview
- ecCharts is based on Metview's architecture and takes it onto the web

Metview and OpenIFS

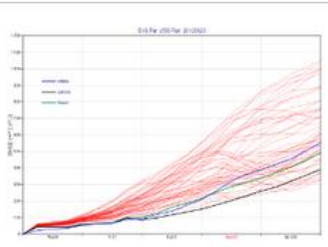
- In previous OpenIFS user workshops case studies were based on custom Metview Macro libraries; now we use Metview Python



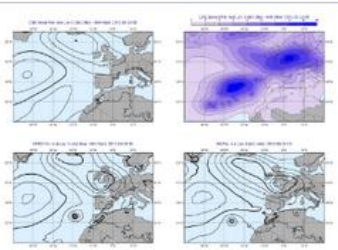
Helsinki, 2013

Toulouse, 2016

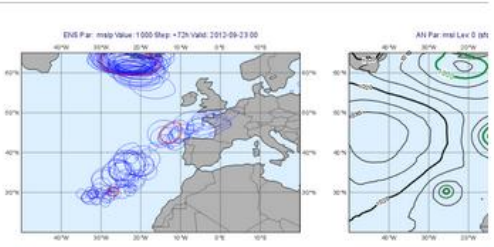
Ensembles



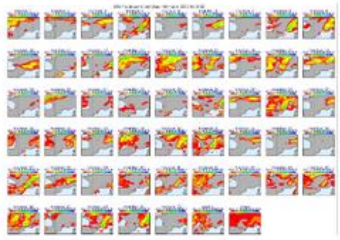
RMSE curves Z500 for ensemble



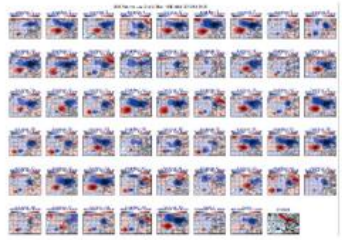
MSLP ensemble mean and spread compared to analysis



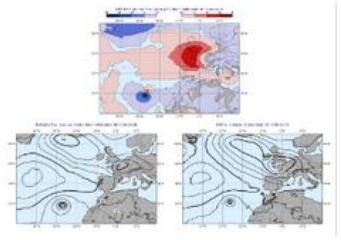
MSLP ensemble spaghetti plot



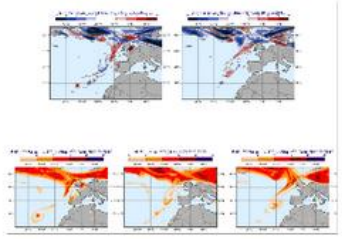
Ensemble stamp map total precipitation over France



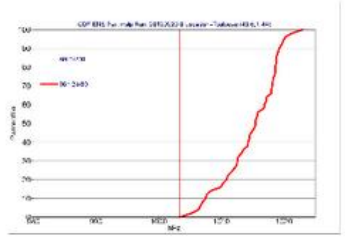
Ensemble difference stamp map of z500



Comparing MSLP of perturbed ensemble membe

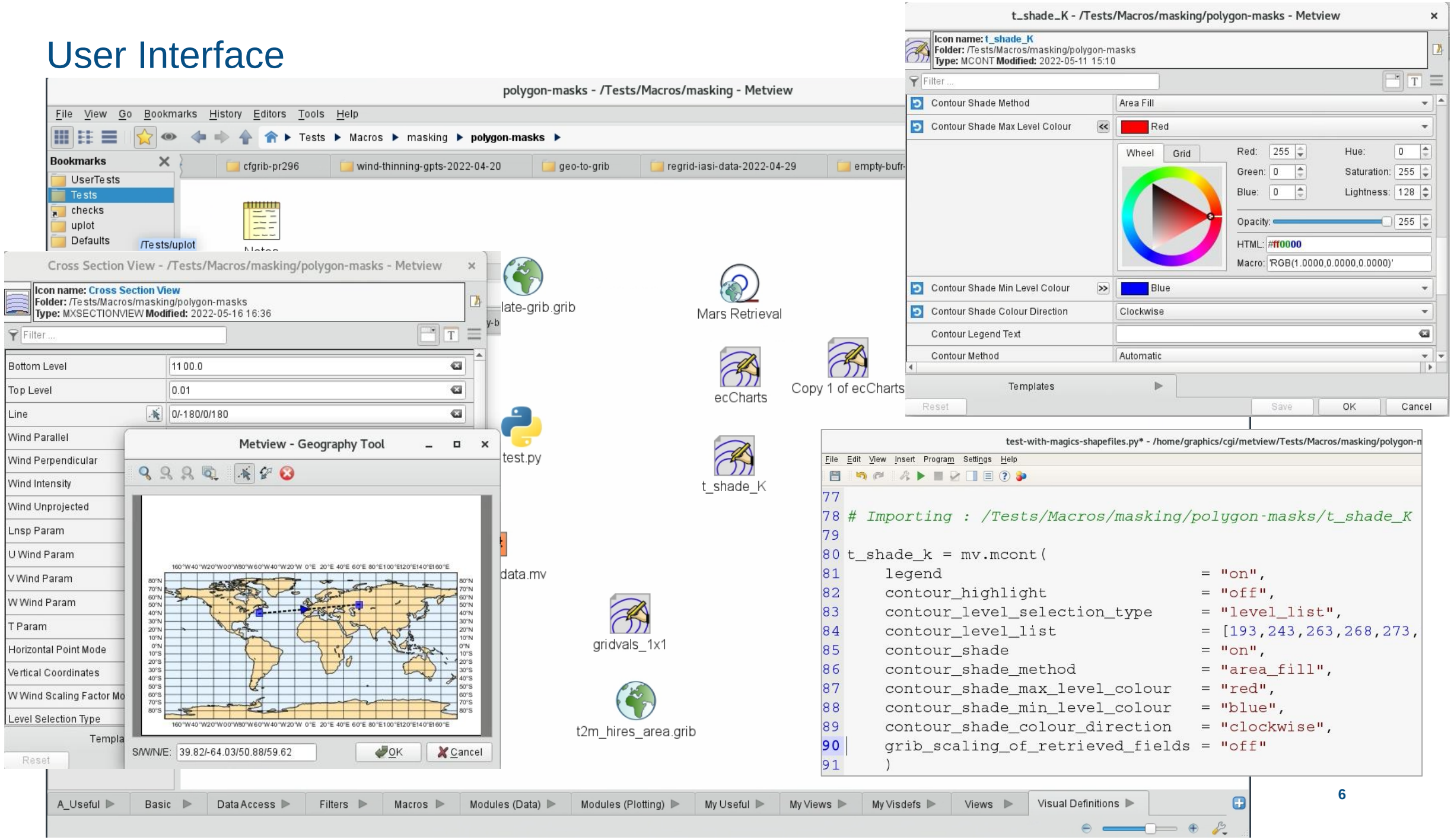


Difference maps of two ensemble members to analysis



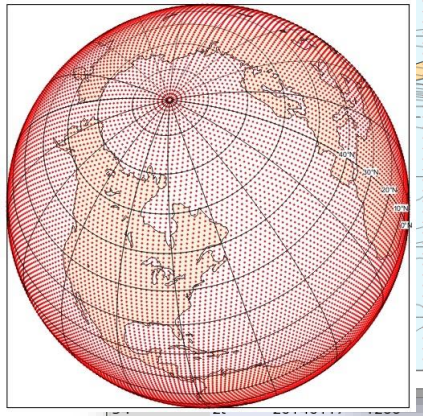
Cumulative distribution function MSLP for ensemble forecast at Toulouse, France

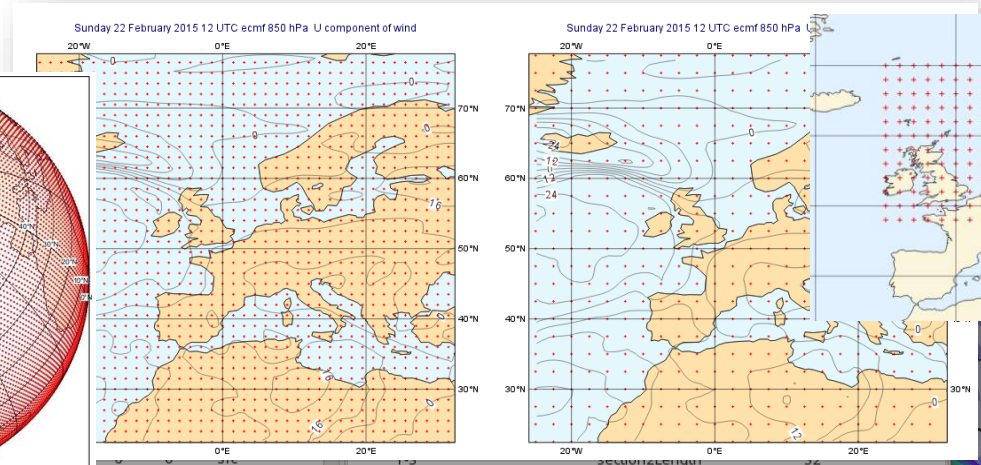
User Interface

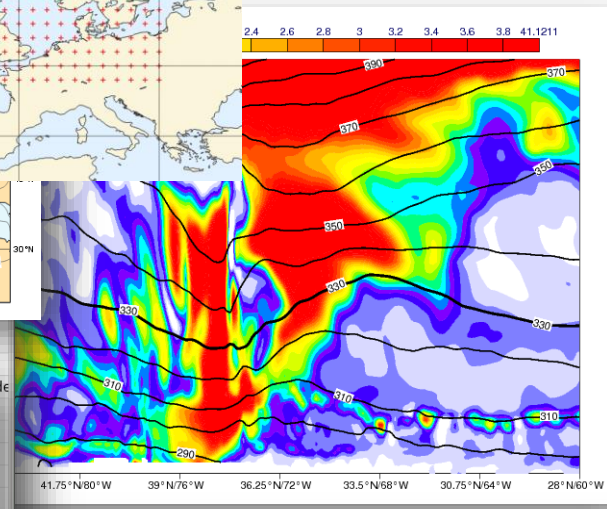


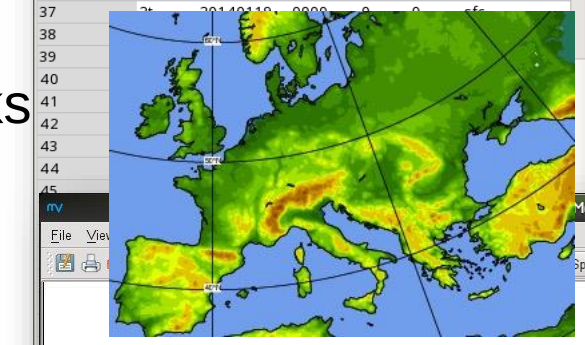
Metview + GRIB

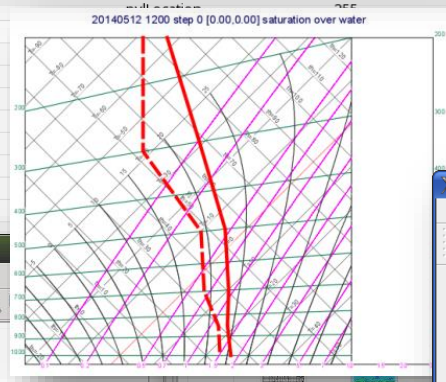
- Plot
- Examine
- Filter
- Spectral transform
- Regridding, cropping
- Missing values, masks
- Maths, Boolean
- Specialised:
  - Cross section
  - Thermodynamics
  - Gradient
  - Vertical integration
  - Model to pressure lev
  - Etc

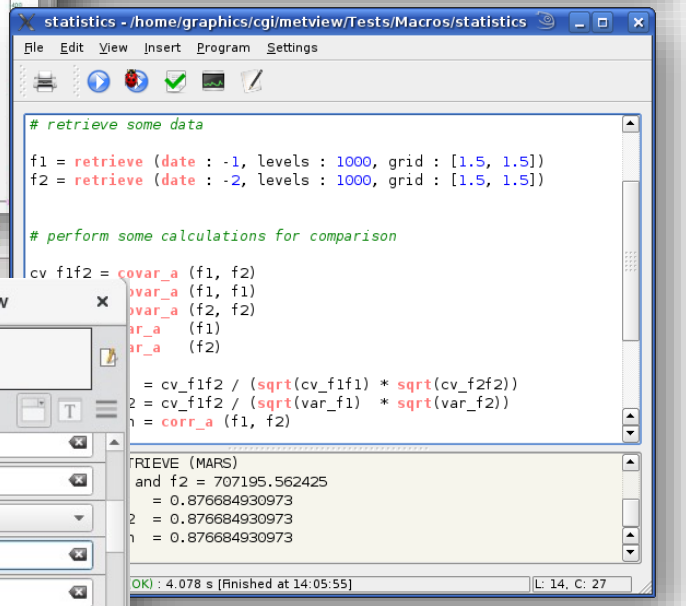


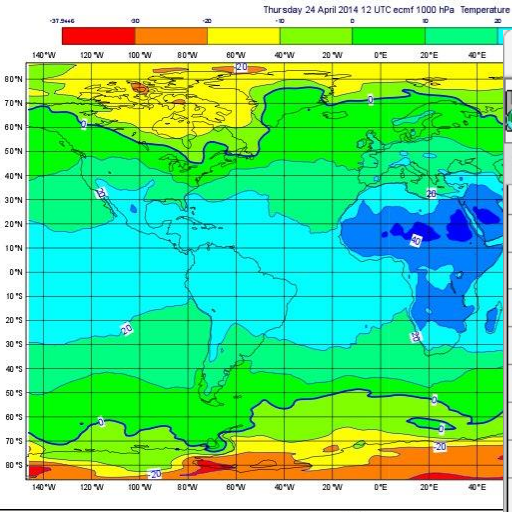






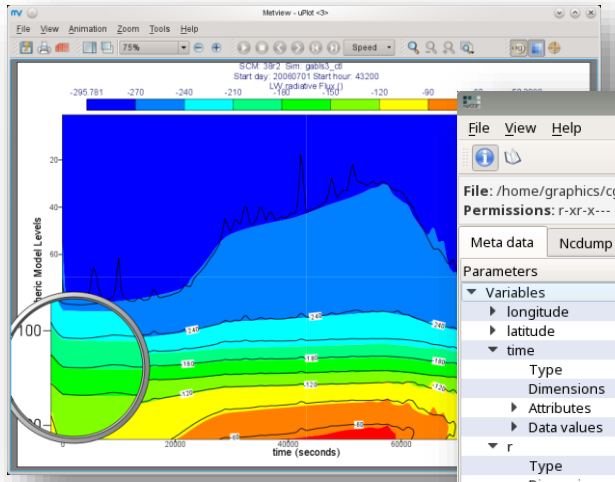


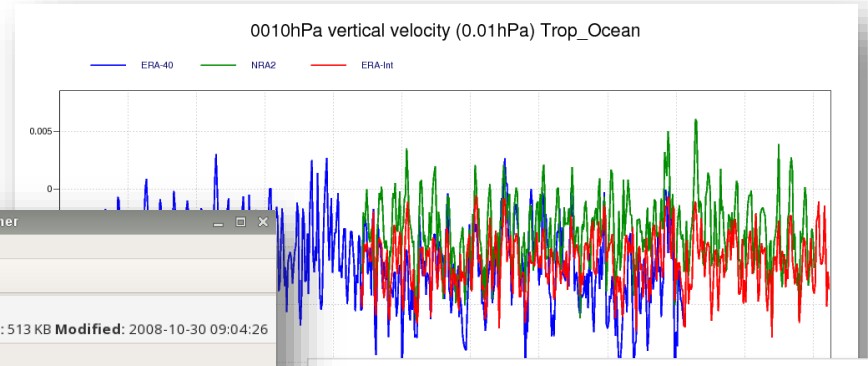


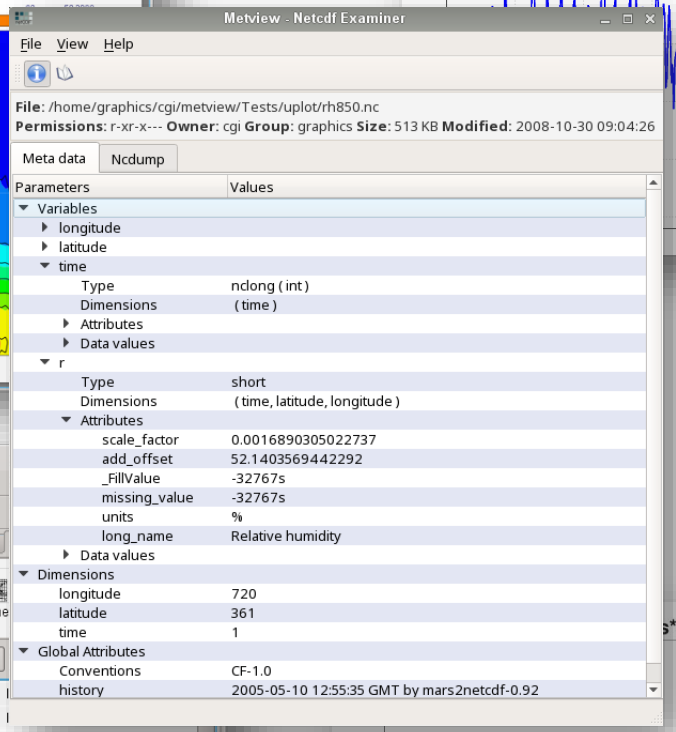


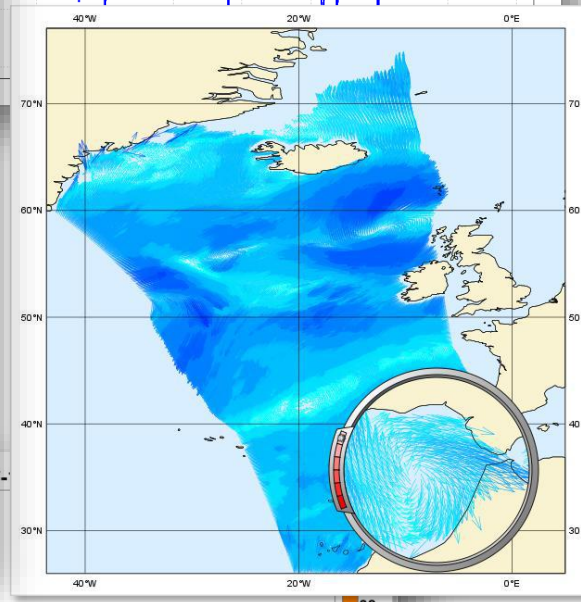
Metview + NetCDF

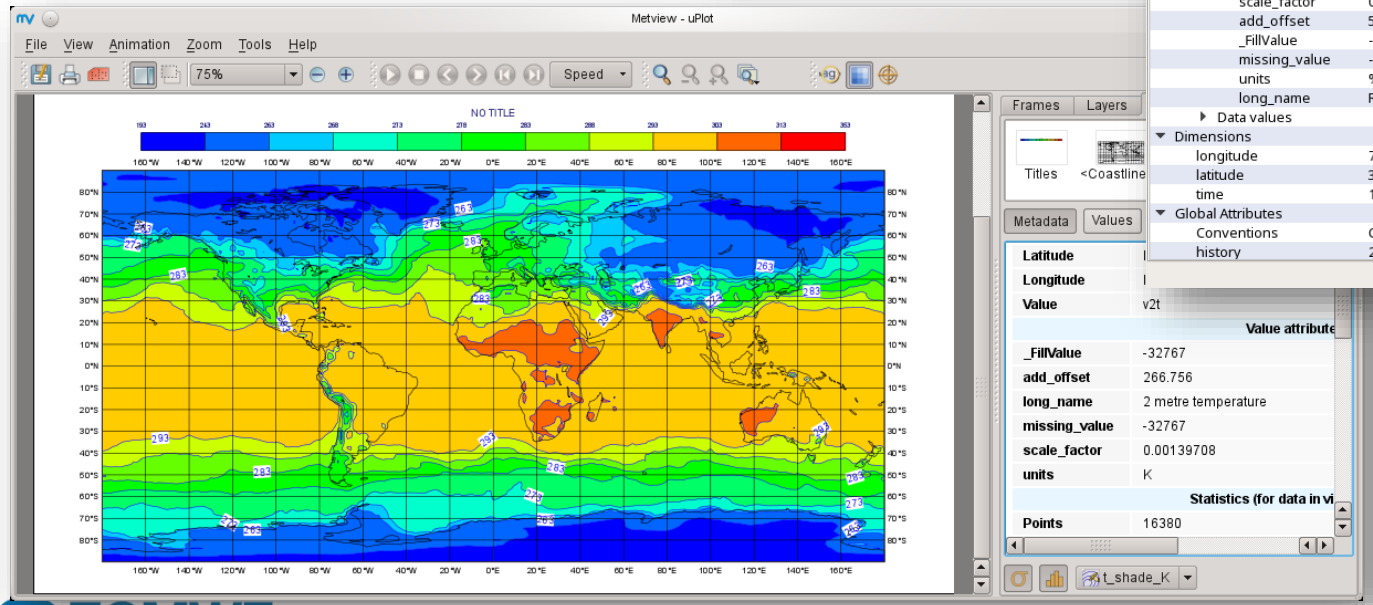
- Plot
- Examine
- Maths, Boolean

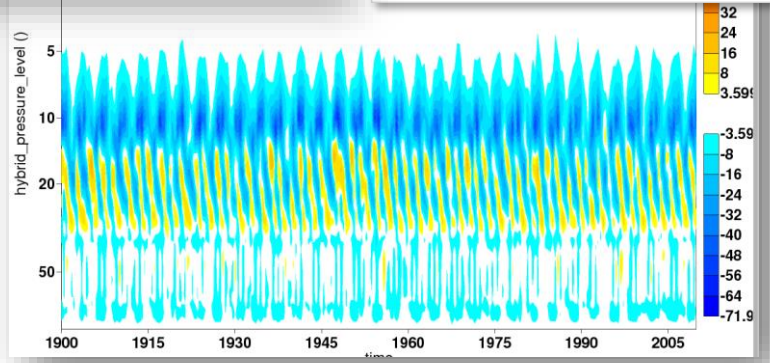






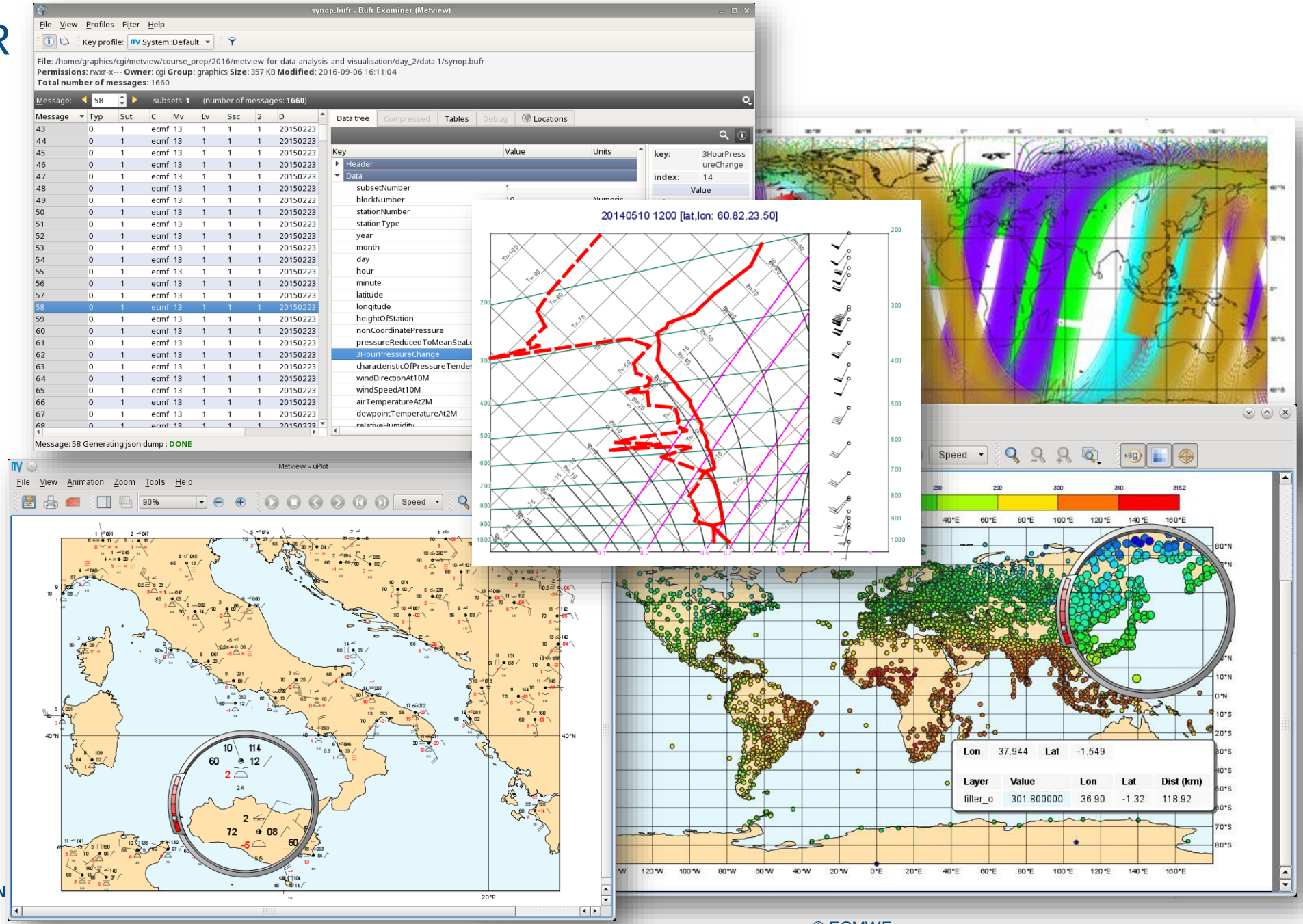






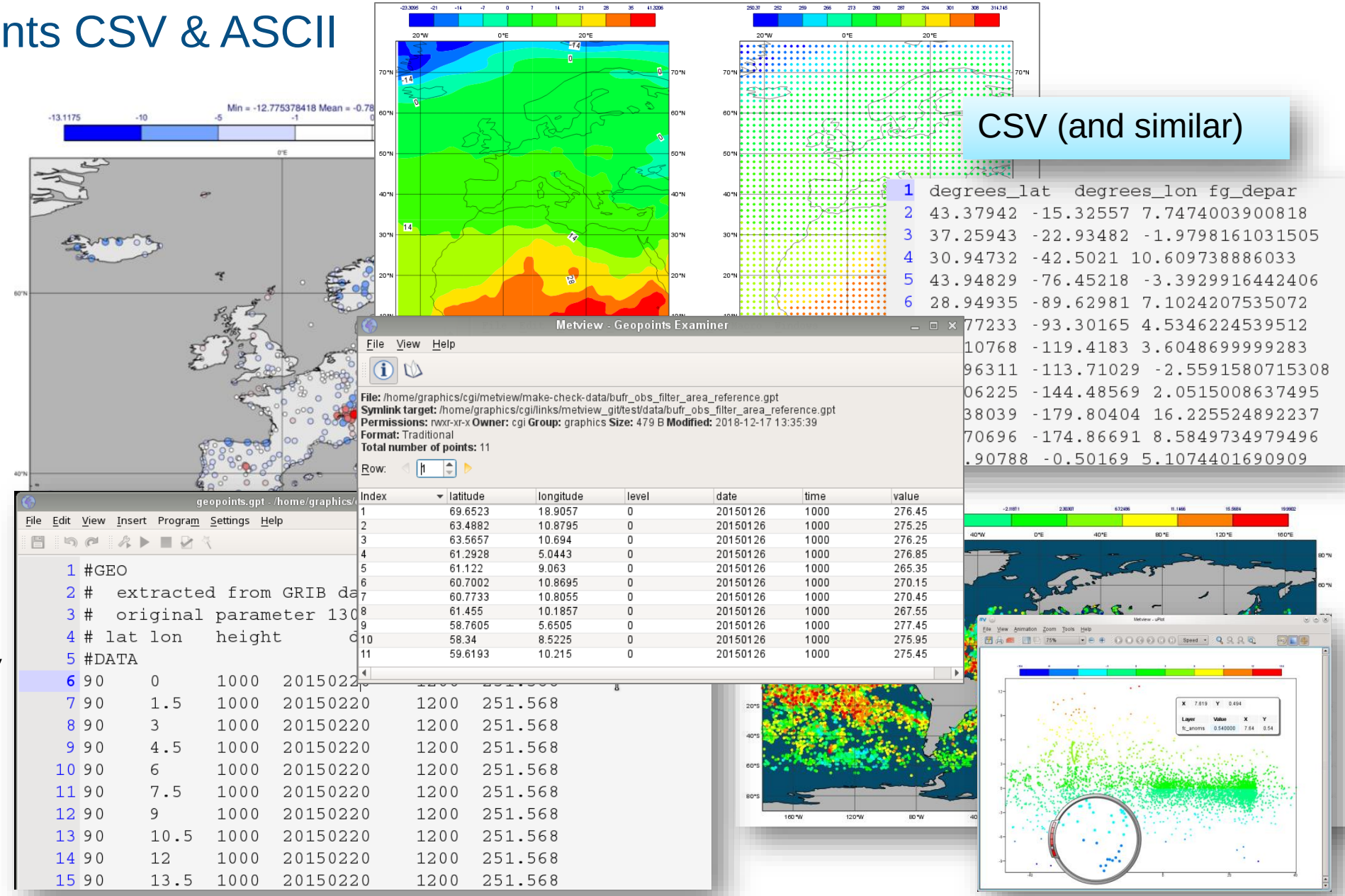
Metview + BUFR

- Plot
- Examine
- Filter
- Extract values
- Convert to Geopoints
- Thermodynamics

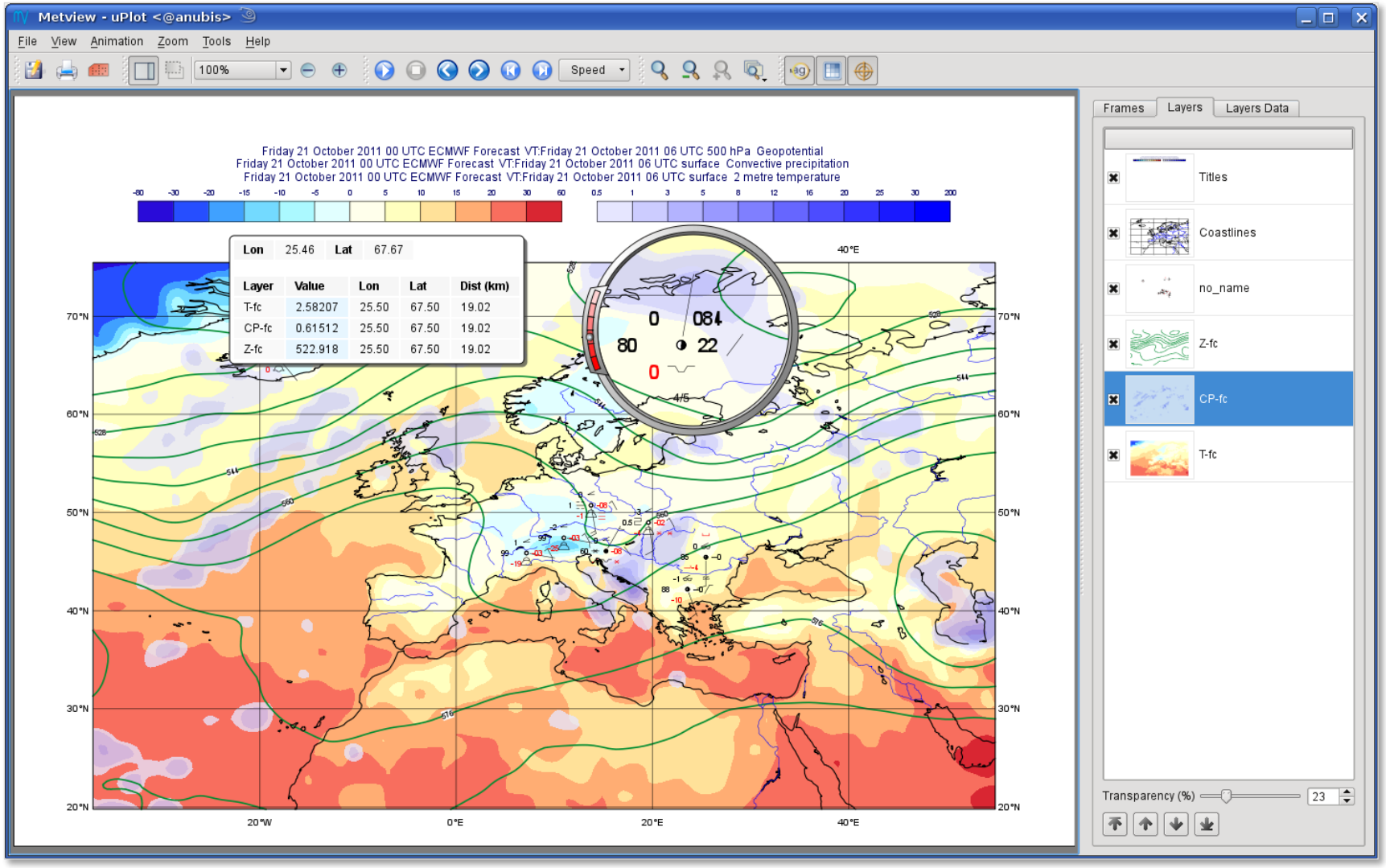


Metview + Geopoints CSV & ASCII

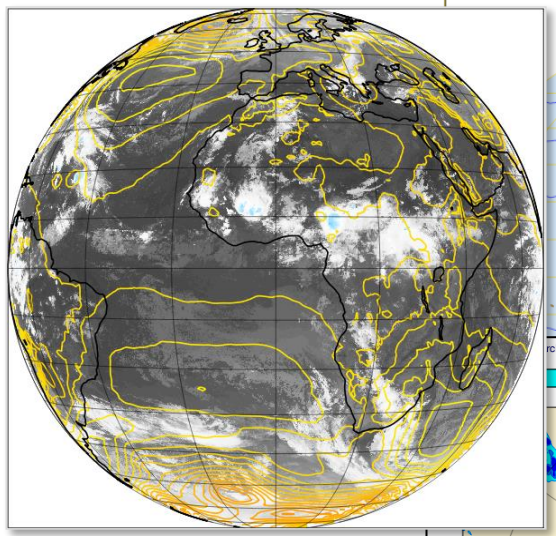
- Geopoints – geo-located values
- Plot
- Examine
- Filter
- Maths, Boolean
- Geo functions
- Convert between GRIB, BUFR and Geopoints
- Can also read CSV

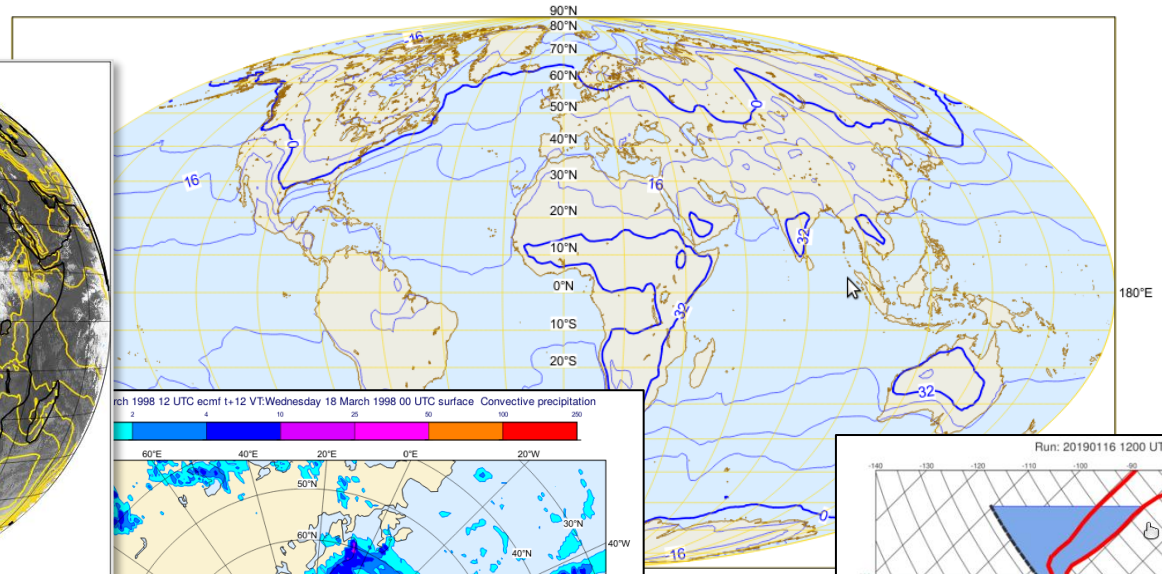


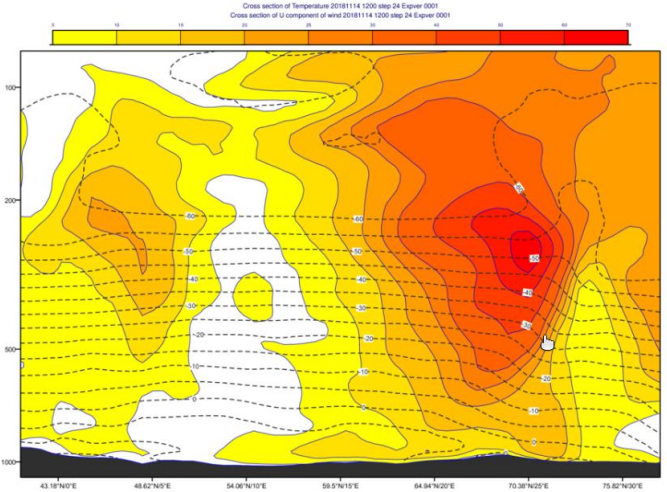
Visualisation - Overlay

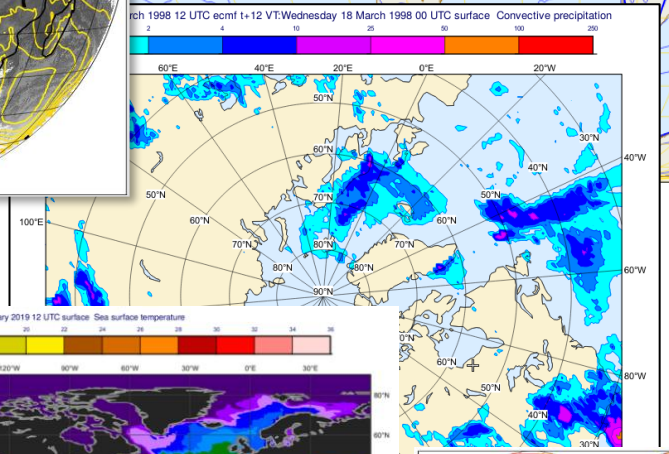


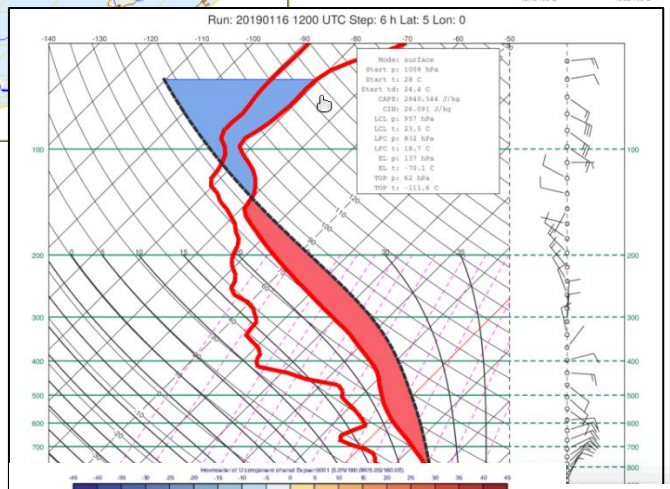
Views

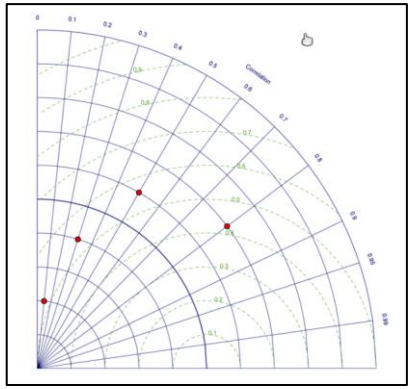


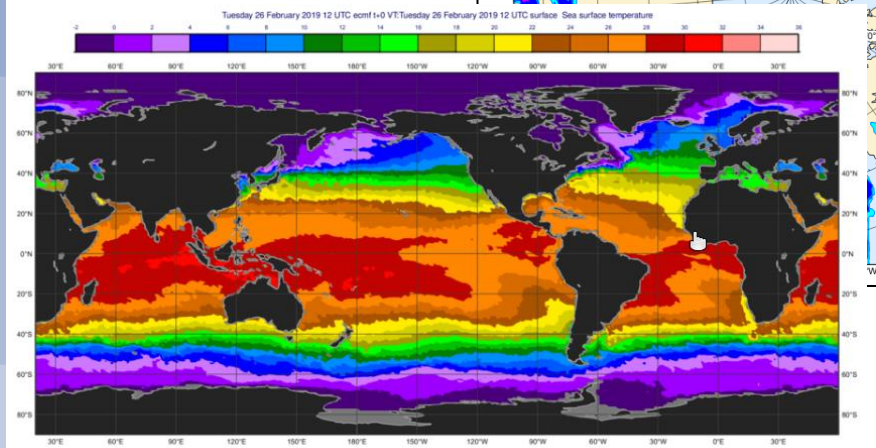


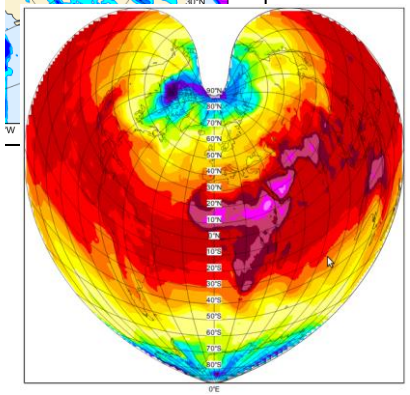


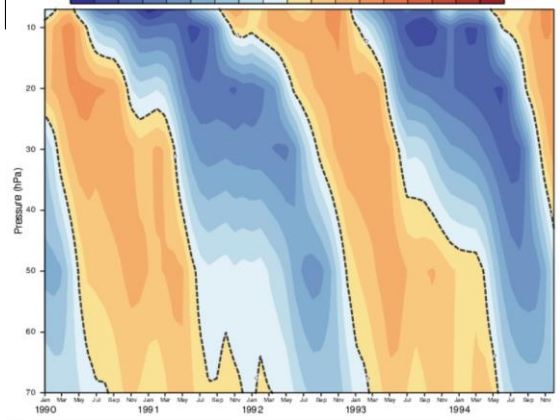


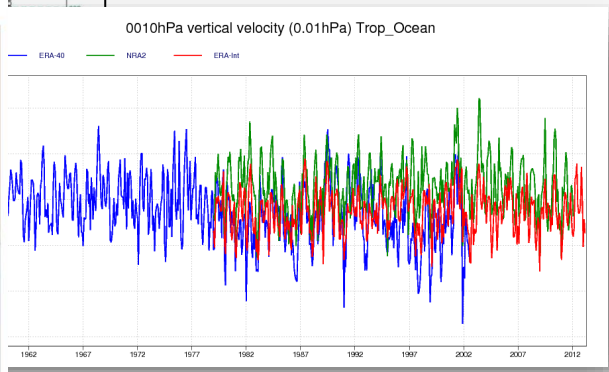










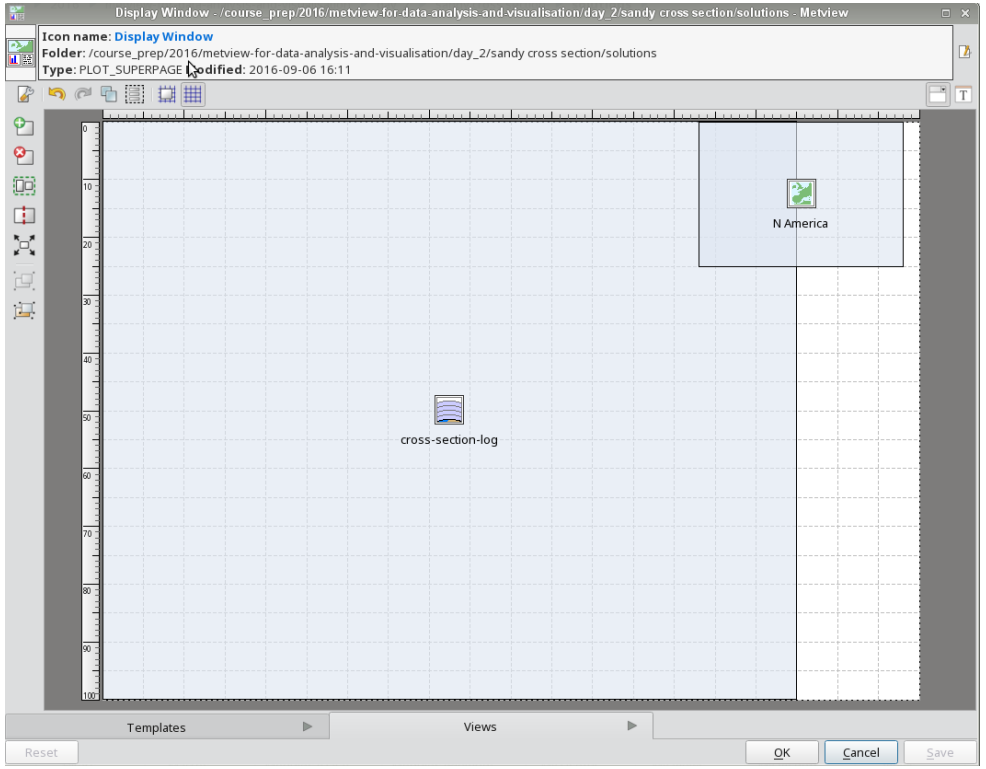


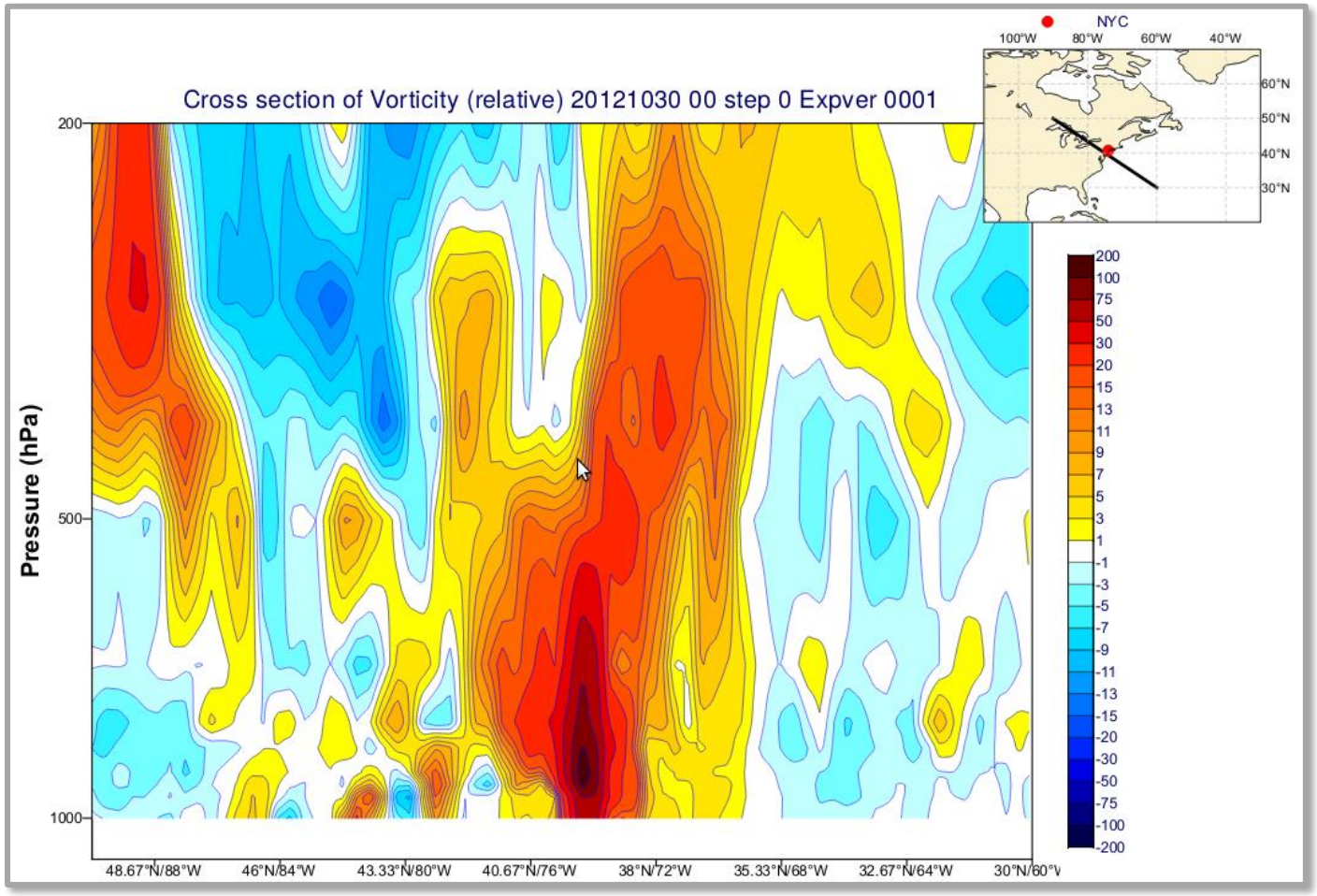
ECMWF

EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

Visualisation - Layout

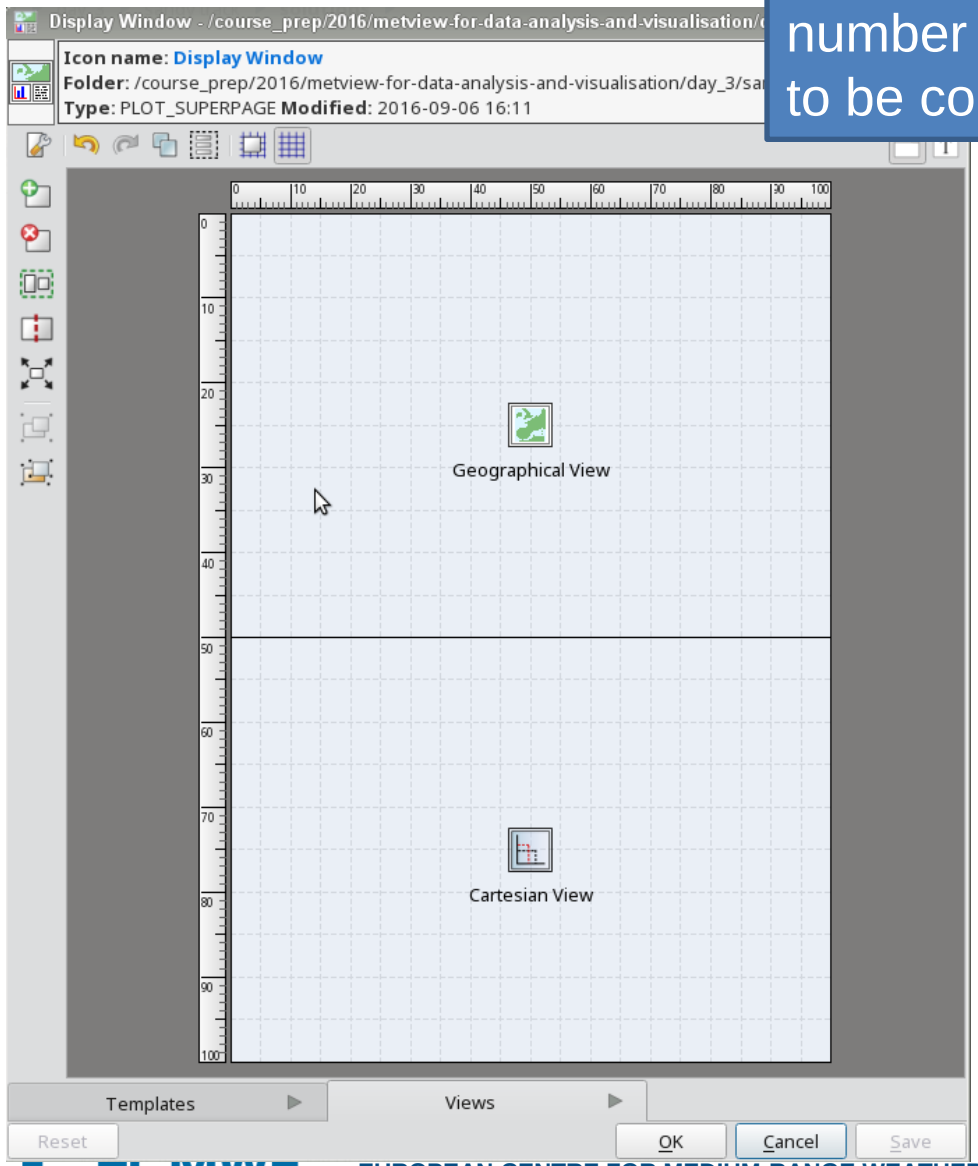
Layout editor allows any number of different views to be combined

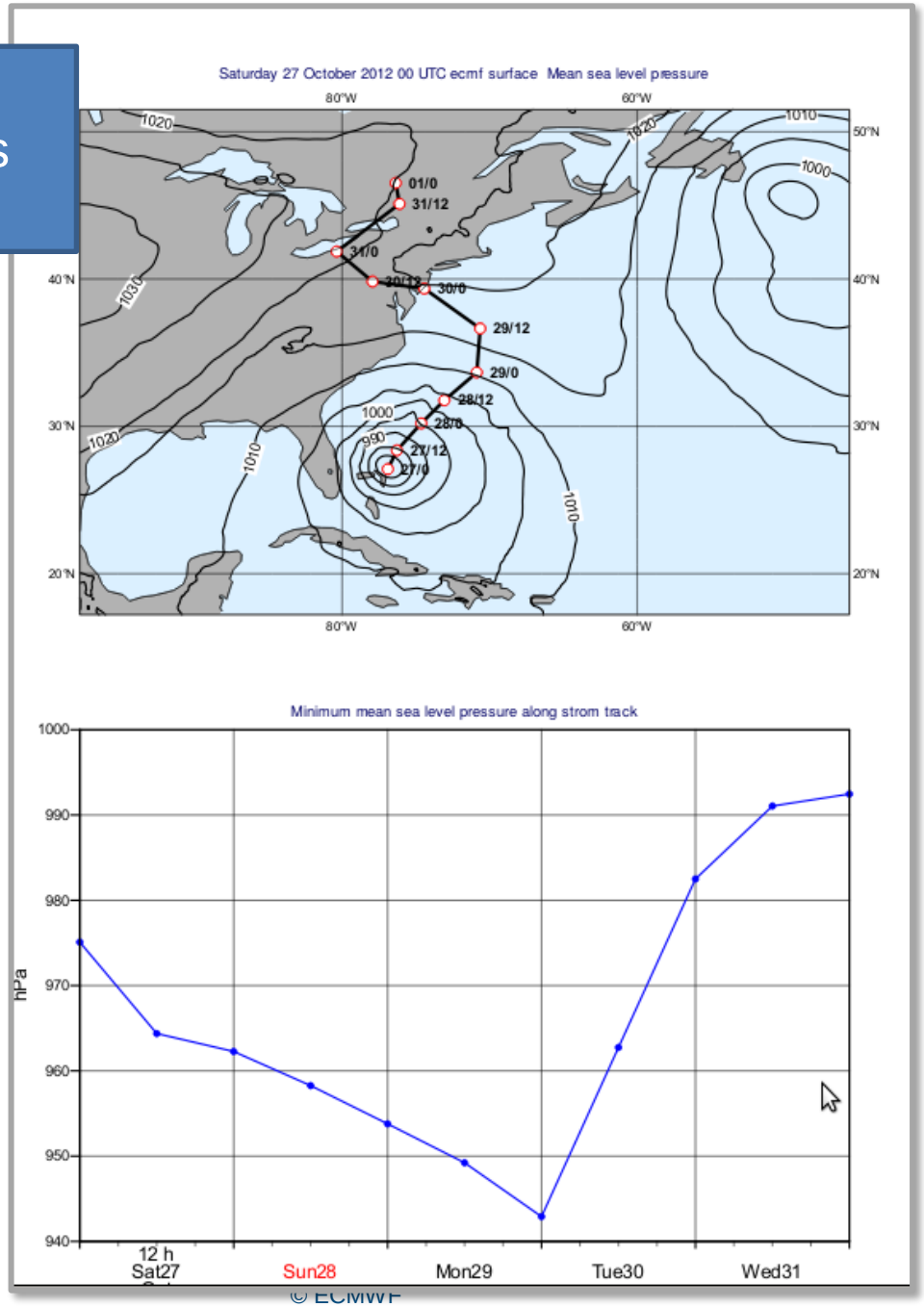




Visualisation - Layout

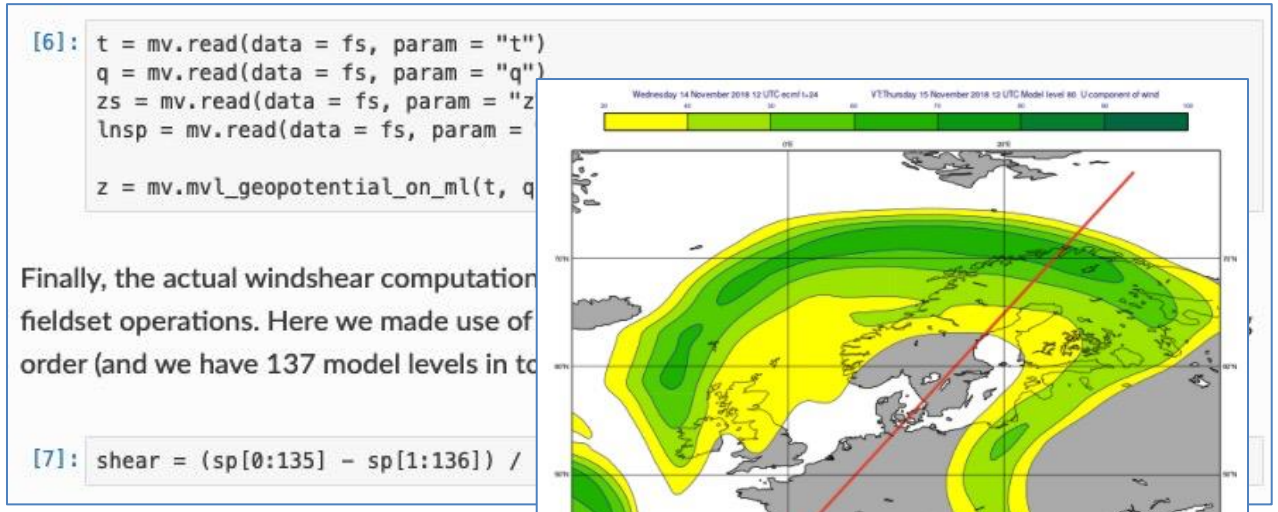
Layout editor allows any number of different views to be combined

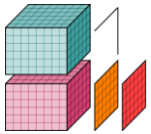




Metview' Python interface

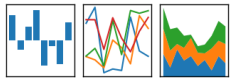
- Gives access to Metview's data retrieval, processing and visualisation capabilities in Python
- GRIB data is loaded as a *Fieldset*
- Can also return data as numpy, pandas and xarray
- Works with the user interface or standalone (UI can even generate Python code for you)
- New features include an interactive plotting widget and data overview functions
- We will use some new helper functions designed to give one-line access to useful plot layouts and styles ; also *datasets* – combination of data and pre-prepared styling

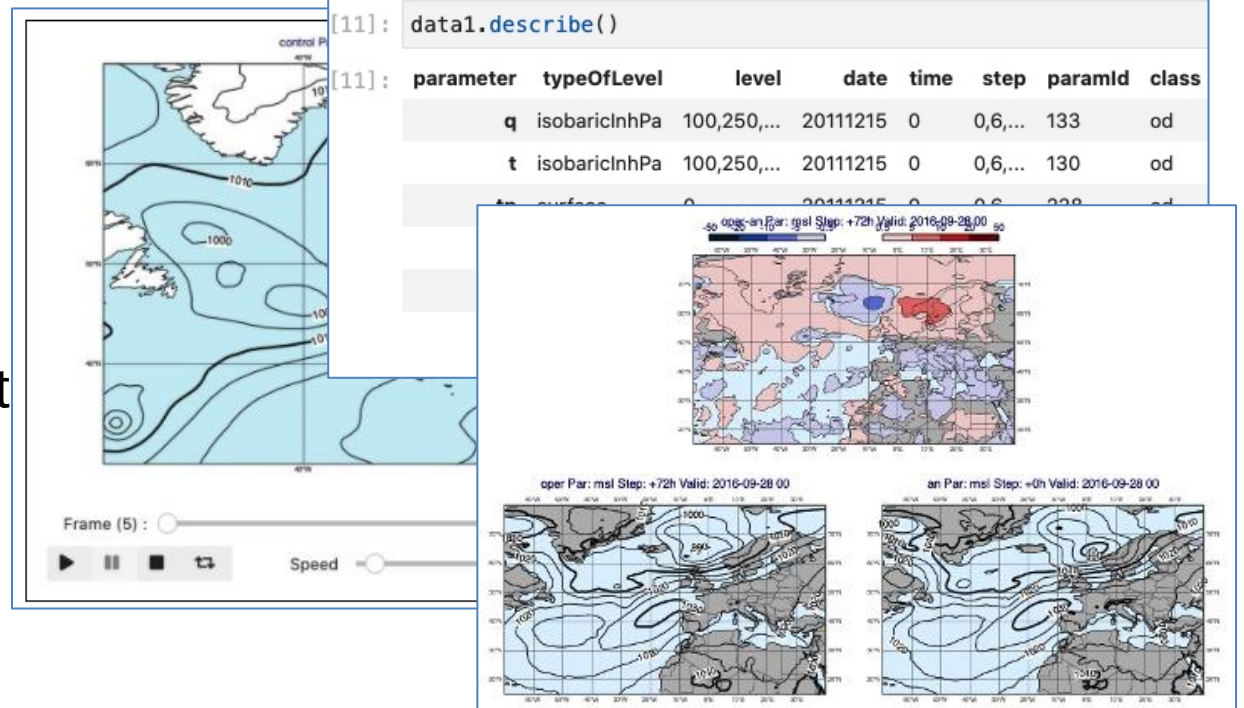




xarray

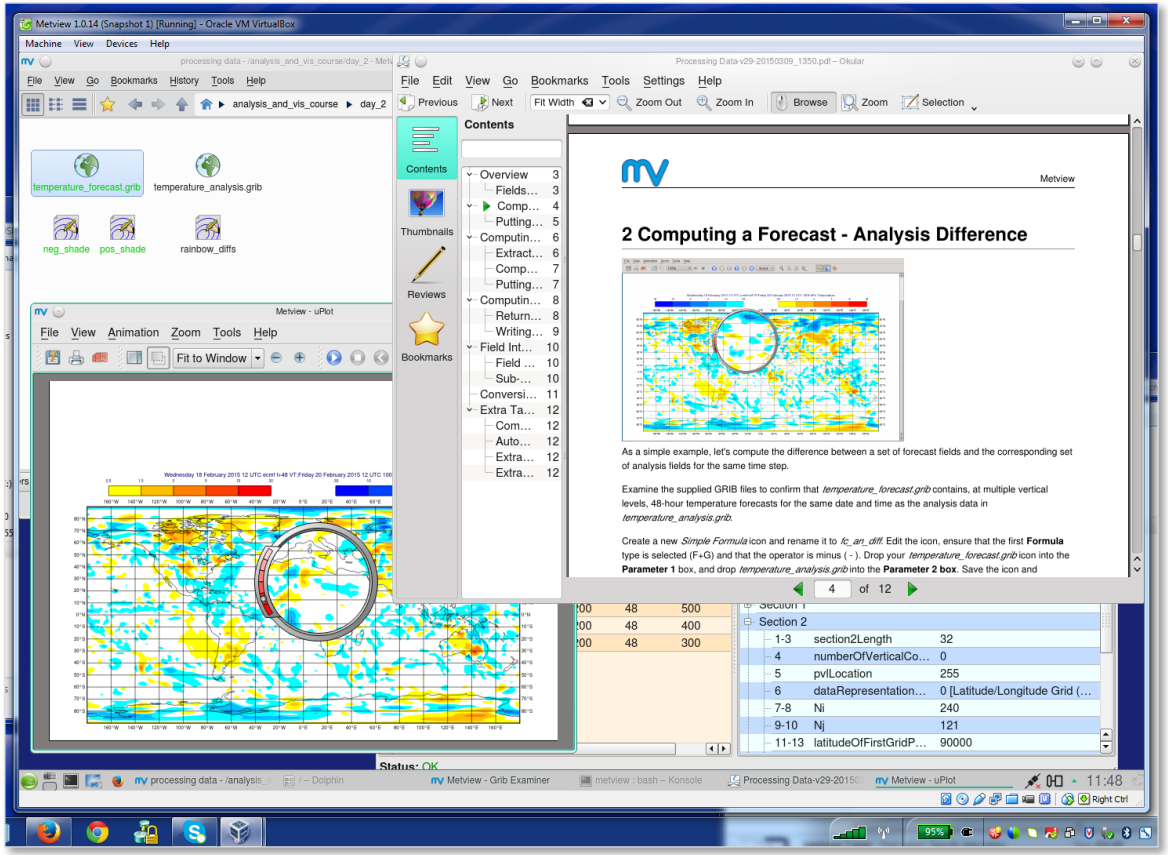
pandas





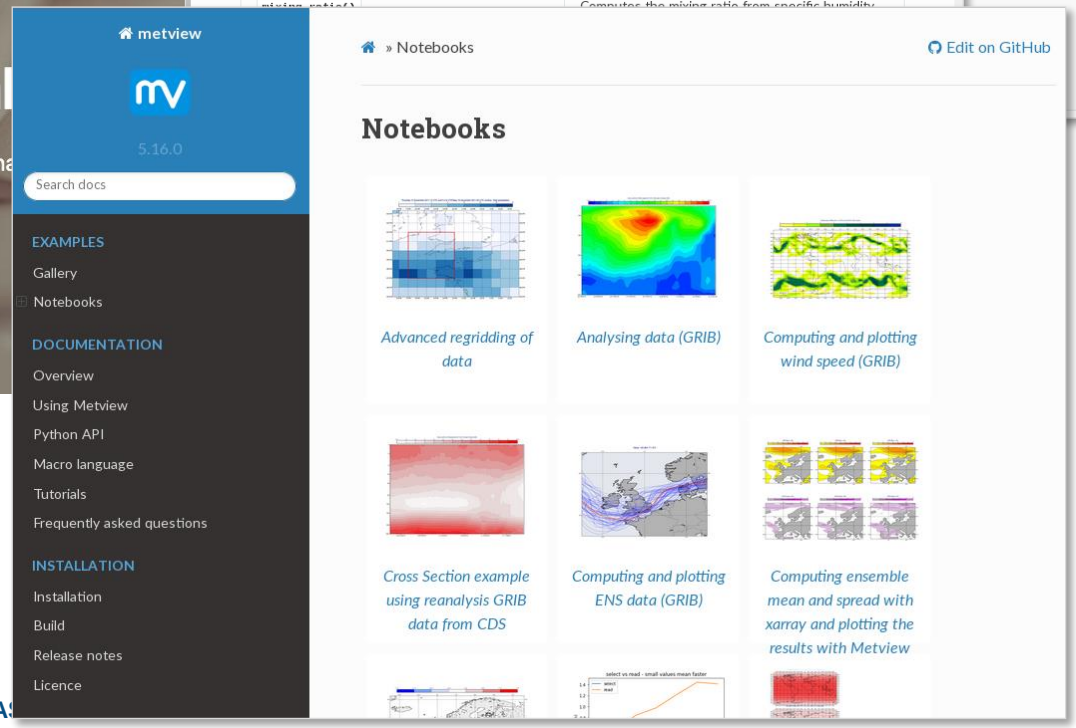
Metview availability

- Available for Linux and macOS
- Inside ECMWF
  - **`module load ecmwf-toolbox ; metview`**
- Install from binaries
  - openSUSE, Fedora, Ubuntu
- Conda
  - **`conda install metview -c conda-forge`**
    `conda install metview-batch -c conda-forge`
    `conda install metview-python -c conda-forge`
- Homebrew
  - **`brew install metview`**
- Build from source
- The Metview Python interface can be installed separately if not in conda:
  - **`pip install metview`**



For more information…

- Ask for help:
  - https://www.ecmwf.int/en/support
- Visit our web pages:
  - https://metview.readthedocs.io/en/latest/index.html



Questions?

Practical session

Inside Jupyterlab browser:
home/Metview_hands_on/Metview_Introduction.ipynb