Developments done during Polly Schmederer's ACCORD VS at DMI (June 2024)

Polly Schmederer (GeoSphere Austria), Carlos Peralta (DMI) and Fabrizio Baordo (DMI)

Topics of VS

Generalising spatial verification:

- Use of reticulate package to interface R with Python.
- Generalise/ apply harp panel tool ("panelification")

  This is a harp and R based version of the originally python based panelification.
- Provide R scripting examples for reading, verification, harp panel tool usage.

Data used for spatial verification

Observations:

- DMI's radar precipitation product: Surface Quantitative Precipitation Estimation (SQPE) using both rain gauge and radar data
- EUMETSAT SEVIRI data (https://api.eumetsat.int/data/browse/collections): High Rate SEVIRI Level 1.5 Image Data - MSG - 0 degree (native), e.g. MSG3-SEVI-MSG15-0100-NA-20240102235743.693000000Z-NA.nat

NWP:

Grib files output of the DEODE workflow running HARMONIE cy46h1 (total precipitation and FULL POS simulated radiances channels WV_062 & IR_108)

As part of the WP6 of the CERISE (Copernicus Climate Change Service Evolution) project, DMI is contributing to the spatial verification of snow cover using the harp verification package. There is a website for the project, if you to add in the README: https://cerise-project.eu/.

Installation instructions

Development was done on ATOS (shared using accord group).

Refer to the installation instructions for details of how to install the libraries.

Overview of the repository

- **reading_functions (in scripts folder)**

  using reticulate

  - `reading_functions.R`: Contains reading functions written in R.
    - read_msg_reticulate() calls python function that reads/ regrids the data and converts the returned data into a harp data frame.
    - read_nc_reticulte() calls python function that reads snow data and converts the returned data into a harp data frame.
  - `reading_functions.py`: Contains the python functions that are called by R to read / regrid satellite observations (.nat) and model data (.grib).
    - sat_model_to_same_grid() reads/ regrids satellite data.
    - get_data_nc_file() reads regridded snow data in nc format.

  using R/harp

  - `reading_functions.R`: Contains reading functions written in R.
    - read_param_with_grbmessg() uses grib message to read simulated satellite channels as they are defined in the file as "unknown".
    - read_deode_tp() adds "tirf", "tgrp" and "tsnowp" to "tp" (total precipitation).
- **panelification**

  in scripts folder

  - `run_panelification`: Run the panelification scripts with set input parameters.
  - `panel_main.R`: Main scripts for panelification. Reads configs, does the verification, calls ranking and plotting functions.
  - `panel_ranking_functions.R`: All functions for ranking the scores are collected in this file.
  - `panel_utils.R`: Some additional functions that are called by panel_ranking_functions.R.
  - `panel_plotting_functions.R`: All functions for plotting the panelification tool can be found here.

in panel_configs folder

   - `panelification.yml`: Sets the configs for which the panel tool shall be run. E.g. date, parameter, models, lead_time and which config files shall be used for the reading of model/obs reading and their verification. Switch that allows to plot FSS and fields separately.
   - `definitions_tp_data.R`: Collects the information needed to read and verify total precipitation of the DEODE experiments against DMI's radar composite.
   - `definitions_sat_data.R`: Collects the information needed to read and verify simulated satellite channels from DEODE experiments agains SEVIRI data.
   - `definitions_tp_plotting.R`: Defines colour scheme and breaks for precipitaion fields.
   - `definitions_sat_plotting.R.`: Defines colour scheme and breaks for infrared fields.

- **data (in sample_data folder)**

  This folder contains sample data which allows to test and run the functionality of spatial verification.

  NWP data in `sample_data/deode/` or `sample_data/dini/`,

  radar precipitation products in `sample_data/radar/`,

  snow data in `sample_data/snow_data/`.

  EUMETSAT SEVIRI data must be downloaded and placed in `sample_data/seviri/`.

- **examples (in scripts folder)**

  read data using reticulate

  `example_read_DataUsingReticulate.R`: Example of how to read satellite data using reticulate. Plots how the output of the functions (directly and via read_grid) looks like. If manually run in a terminal one can also use the plotting example to see how the fields actually look.

  spatial verification

  examples on how verify_spatial can be used used:

   - `example_verify_tp_deode.R`: An example file for verifying total precipitation of DEODE experiments.
   - `example_verify_sat_deode.R`: An example file for verifying simulated brightness temperatures against SEVIRI data.

    - `example_verify_snow_cover.R`: An example file for verifying regridded snow data.

How to...

... run the panelification

- Run panel script `./scripts/run_panelification` to see whether the scripts and installation works for you.
- Check `run_panelification` to see how to call the panelification main function `panel_main()`.
- To

    - run *different dates* or *lead_times* (for which the mod/obs data is available),
    - decide *which models* should be displayed or
    - switch separate *plotting of FSS and fields* ON or OFF

  change the settings accordingly in `panel_configs/panelification.yml`.

- To run panelification out of the box

    1. cd ACCORD_VS_202406
    2. a) on ATOS:

       load modules:

       ```
         module load R/4.3.3
         module load ecmwf-toolbox/2024.04.0.0
         module load hdf5/1.14.3
         module load proj/9.3.1
         module load python3/3.11.8-01 (only necessary
       to run sat_verif)
       ```

       b) on virtual machine:

       ```
         conda activate <conda_env>
       ```

    3. export your R local installation, e.g.: (in case step ii alone is not working)

```
export R_LIBS_USER=<path_to_harp_local_
installation>/renv/library/R-4.3/x86_
64-pc-linux-gnu
```

4. run panelification for precipitation, e.g.:

```
Rscript ./scripts/panel_main.R prec_verif
```

Plots, as a results of point 4, are saved in `PLOTS/` folder.

You can also test the spatial verification for **satellite radiances**. To do that,

firstly, you need to download the expected EUMETSAT SEVIRI data (as configured in panel_configs/definitions_sat_data.R),

secondly, you must replace the python_version in the same definition file to the path where your python is installed. If you are not sure where this is, run in your terminal:

```
which python
```

then you can run:

```
Rscript ./scripts/panel_main.R sat_verif
```

... interprete a panelification plot

The scores, which are defined in the definition file, are calculated using harpSpatial.

The models are then ranked according to the scores.

The ranks are plotted in the panelification plots.

Colour scheme of the ranks:

- perfect scores (green)
- rank 1 (gold)
- rank 2 (silver)
- rank 3 (bronze)
- rank 4 and greater (white)
- FSS not skilful (< 0.5) (red)
- NA (black)

When 2 or more models have the same rank, the next model with a higher rank will have a rank that is following up after the "missing" ranks. e.g. three models have rank 2 -> 3 and 4 are the "missing" ranks and the next-ranked model will have rank 5.

Information displayed in the plot

In the first panel the observation field is displayed, there is one panel for each of the verified models.

observations:

obs title: name + valid observation time

models:

- model title:
    - left: model name, initialisation time + lead time, (average FSS rank)
    - right: average rank of basic scores (this is the average of all non-FSS scores that are passed in the definition file) and ranking of the models according to this average rank.
- top box:
    - ranks of FSS (using thresholds)
    - ranks of FSS (using percentiles)
    - basic scores - displaying the actual values (rank according to the value)

For more information on what you see on a panel plot, check the interpretation help.

... add other models

To add another model (panel) to panelification, a file with their definitions on how to read/verify must be added, which will then be added to the panelification.yml.

1. Know how to read the model that should be added. (An example on how the data may be read can be found in `example_read_DataUsingReticulate.R`).
2. Run verify_spatial, to see how the configurations must be passed to this function (see the examples `example_verify_tp_deode.R`, `example_verify_sat_deode.R` or `example_verify_snow_cover.R`).

3. Copy a definitions_* file (`definitions_<new>.R`) and change all parameters as needed to read/verify the data correctly.
4. Specify the new file in `panel_configs/panelification.yml` for model and parameter. (Define which config files to use > READING of the models/obs).
5. If a new parameter was added, also add a definition file for this parameter in `panel_configs/panelification.yml` (Define which config files to use > PLOTTING of the fields).

Example for other models

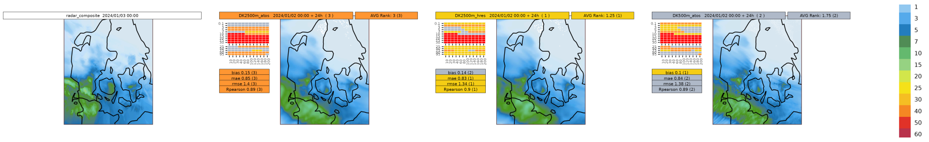
An example using data from the dini versus the ifs model is also provided, with the definitions in `definitions_tp_data_dini.R` . To test this example use the config file in `panel_configs/panel_dini_ifs.yml` and run

```
Rscript ./scripts/panel_main.R prec_verif panel_
configs/panel_dini.yml
```

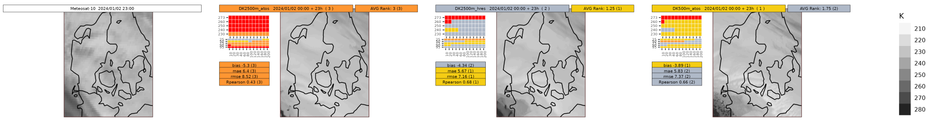
Note that the ifs model data contains the total precipitation field multiplied by a factor of 1e3 to make it comparable with the data from dini, that produces total precipitation in units of `kg m^2`, while ifs outputs data in `m`. The units in the ifs files are not updated. A sample mars script to fetch the ifs data is included.

Some example plots

Panelification plot precipitation



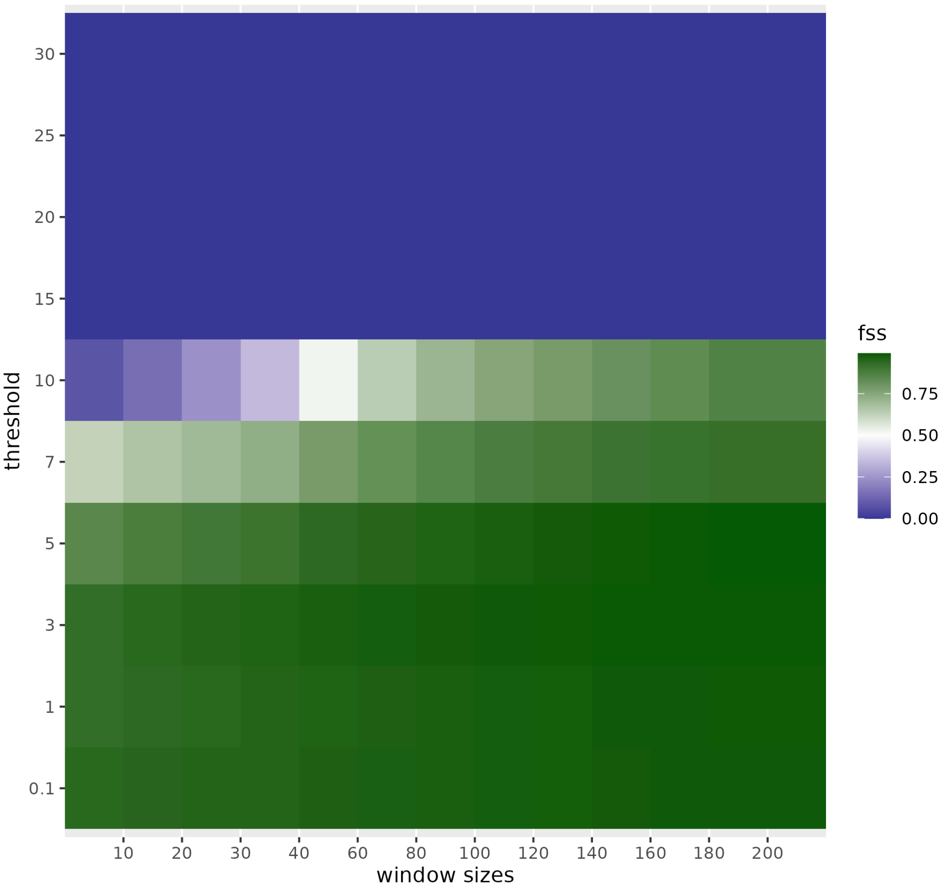
Panelification plot IR 108



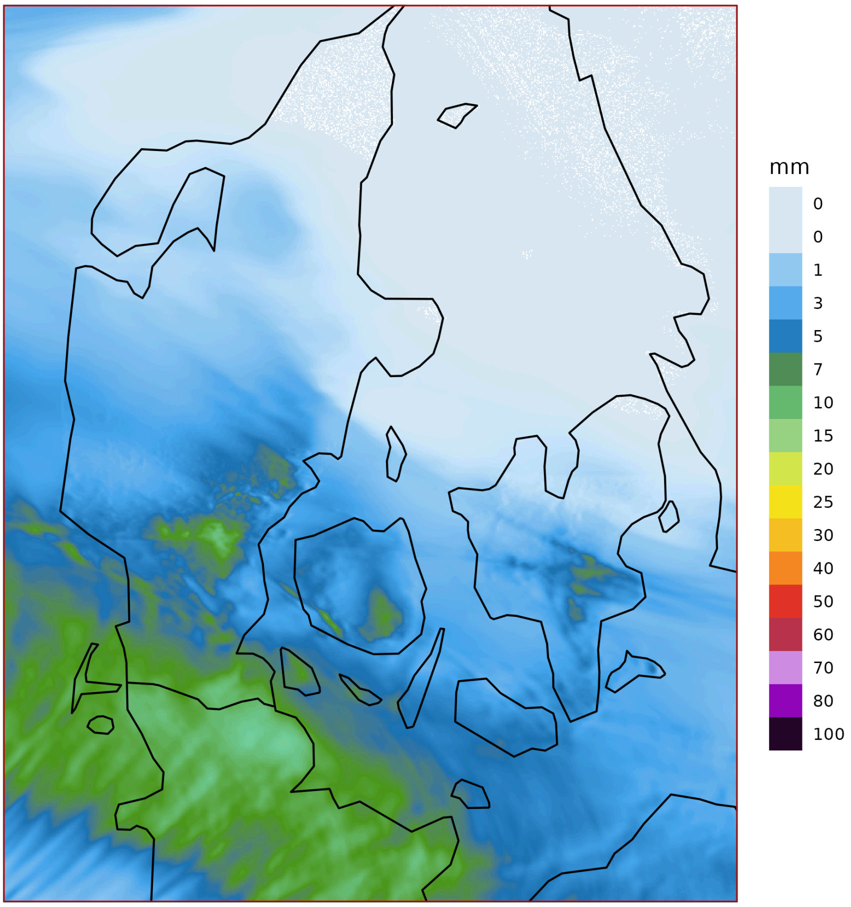
FSS plot precipitation

Model: DK500m_atos , Param: Accpcp3h

Period: 2024010200 + 24



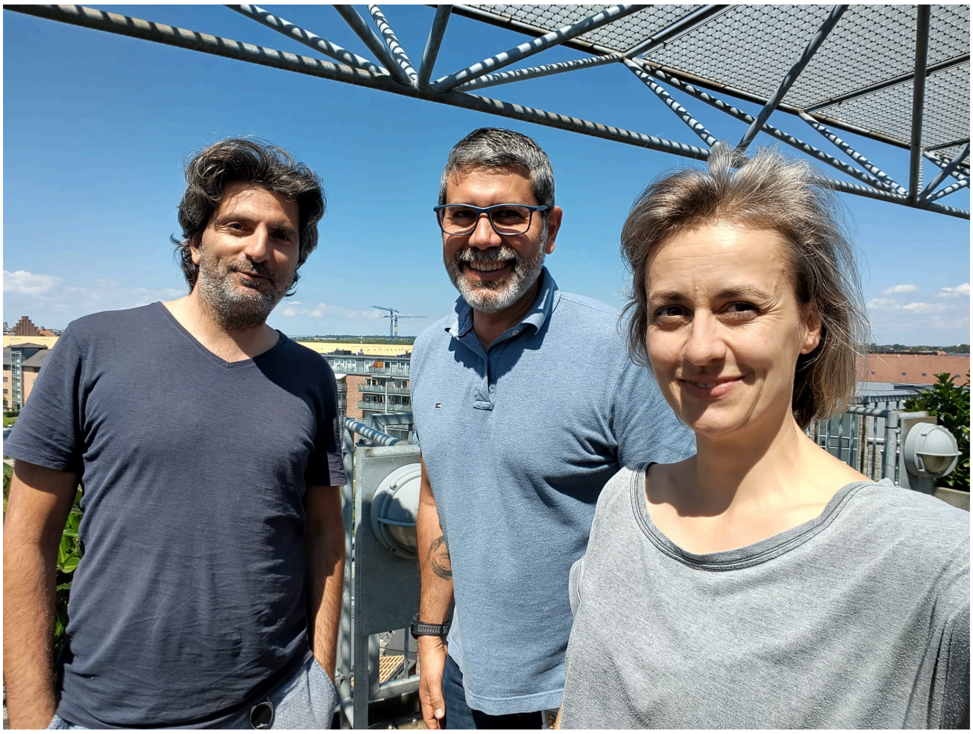
precipitation field



Model: DK500m_atos , Param: Accpcp3h

Period: 2024010200 + 24

Last day of successful stay



Outlook

- Open pull request for https://github.com/harphub/oper-harp-verif.
- Implementing a more sofisticated approach for averaging the FSS ranks: weighting the FSS values according to thresholds / window sizes, in order to give smaller window sizes and higher thresholds higher credit as they are more difficult to forecast.
- Do further investigation of python/reticulate to possibly add new/different gridded data sources.
- Adding score D90.
- Add more plotting options as available in panelification.
- Allow looking for smaller threshold instead of greater threshold (harpSpatial).
- Replace fobs=0.5 by calculated fobs (harpSpatial).
- Allow distinction between verification and plotting domain (harpSpatial).