pysteps - a Community-Driven Open-Source Library for Precipitation Nowcasting

S. Pulkkinen[1,2], D. Nerini[3,4], A. Pérez Hortal[5], C. Velasco-Forero[6], A. Seed[6], U. Germann[3] and L. Foresti[3]

[1]Colorado State University, Fort Collins, United States
[2]Finnish Meteorological Institute, Helsinki, Finland
[3]Federal Office of Meteorology and Climatology MeteoSwiss, Locarno-Monti, Switzerland
[4]Institute for Atmospheric and Climate Science, ETH Zurich, Switzerland
[5]Department of Atmospheric and Oceanic Sciences, McGill University, Montreal, Canada
[6]Bureau of Meteorology, Melbourne, Australia

Motivation

- The existing radar-based nowcasting software packages are proprietary and being developed in national weather services.
- The aim of the pysteps initiative is to bring the nowcasting knowledge available to the public in a Python-based open-source software package.

The STEPS Model
(Bowler et al. 2006)

- The underlying model is Lagrangian persistence.
- The precipitation field is extrapolated using the estimated advection velocities (Fig. 1).
- The model takes into account scale-dependence of predictability.
- Precipitation fields are decomposed into multiple spatial scales using a set of Fourier bandpass filters (Figs. 2 and 3).
- Temporal evolution of precipitation intensities is modeled by using an autoregressive AR(2) model in Lagrangian coordinates (Fig. 4).

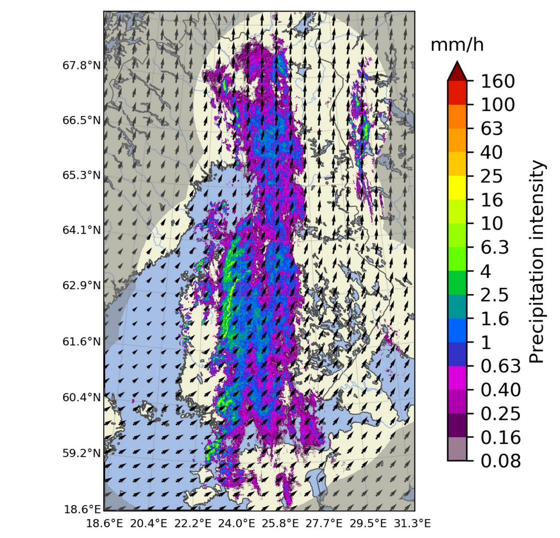


Figure 1. The advection field estimated by using the Lucas-Kanade method with interpolation applied to areas of no precipitation

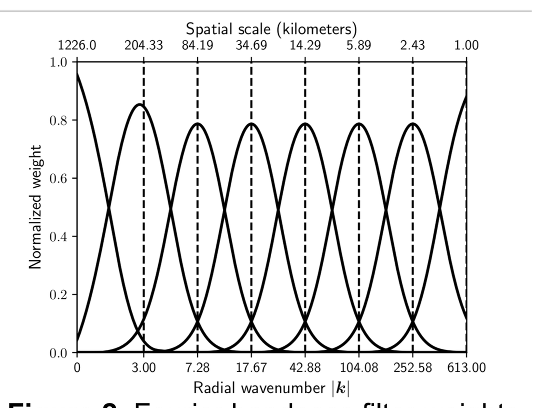


Figure 2. Fourier bandpass filter weights.

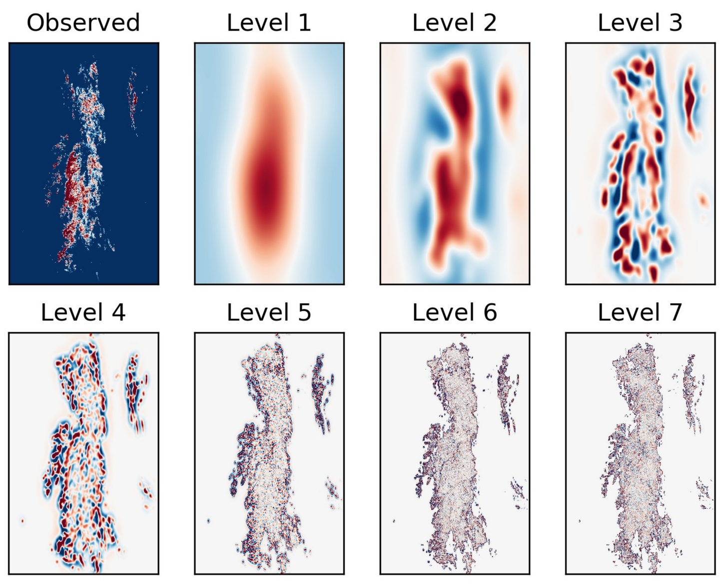


Figure 3. 7-level cascade decomposition of a radar composite.

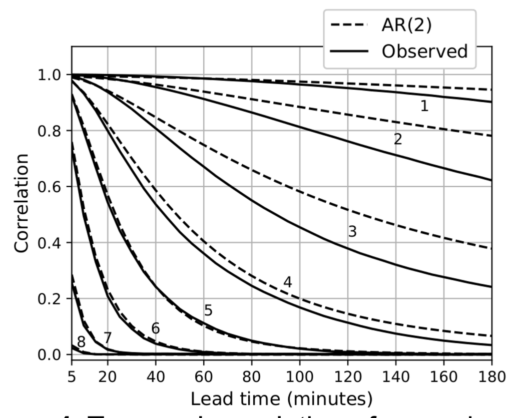


Figure 4. Temporal correlation of cascade levels obtained from 1) the AR(2) model and 2) comparison with verifying observations.

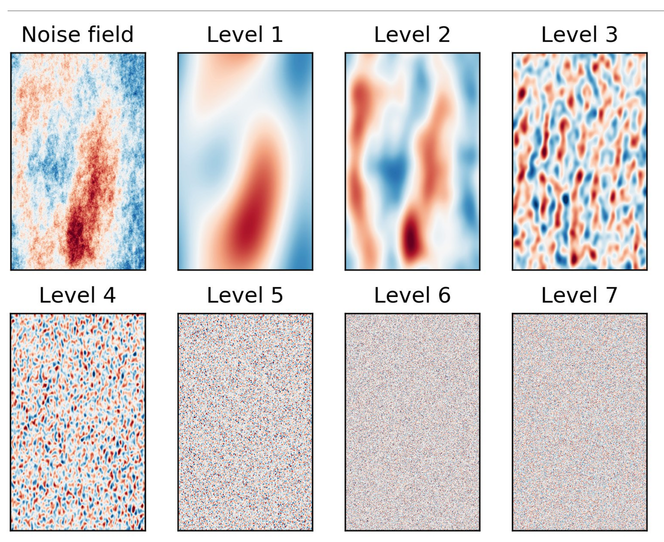
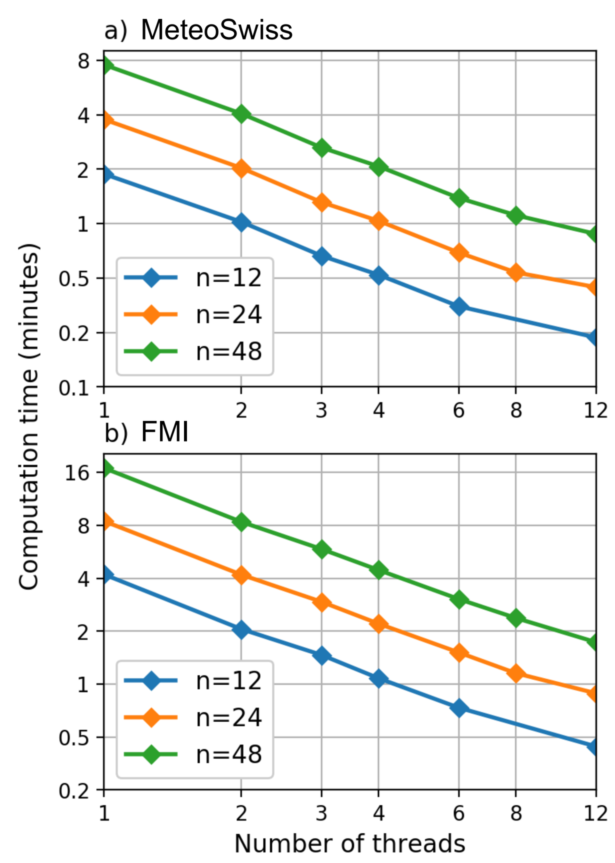


Figure 5. Cascade decomposition of a Fourier-filtered Gaussian random field.

- Stochastic perturbations are added to the precipitation intensity and advection fields to simulate uncertainties.
- For the precipitation intensities, a filtered Gaussian noise field is added separately to each cascade level to account for the scale-dependence of predictability (Fig. 5).

Computational requirements

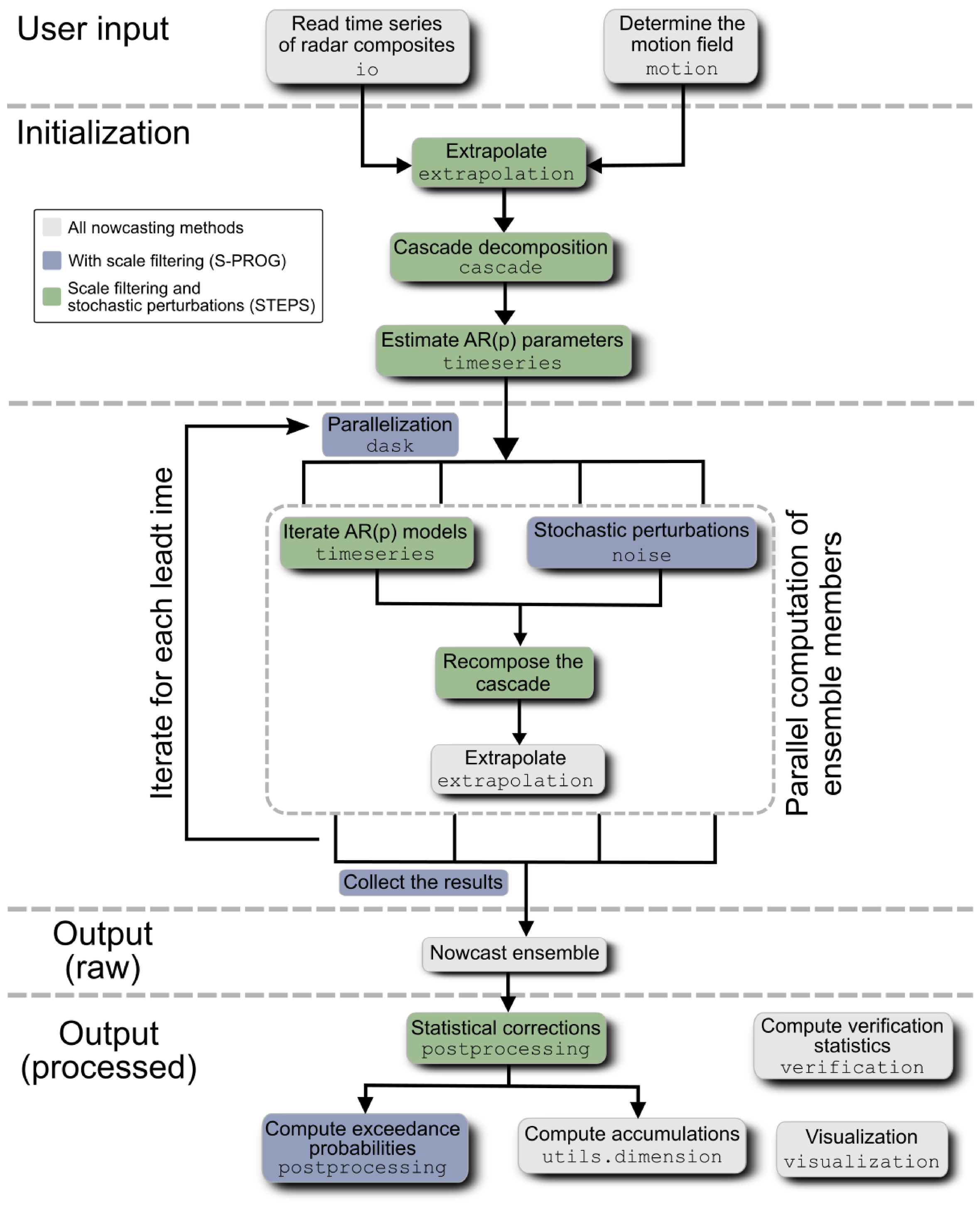
- Computation times were measured using a typical server (2.6 GHz Intel Xeon processor with 12 cores and 64 Gb memory).
- In the MeteoSwiss domain (grid size 710x640 pixels), a 48-member STEPS nowcast ensemble can be generated in one minute.
- In the FMI domain (760x1226 pixels), a 48-member ensemble can be generated in 2 minutes.
- When plotted on log-log scale, the computation time is linearly reduced with respect to the number of processor cores.



pysteps architecture

- **Modular architecture:** the main components of nowcasting methods are interchangeable.
- **Nowcasts with different levels of complexity**:
  simple extrapolation
  ↓
  extrapolation + **scale filtering**
  ↓
  extrapolation + scale filtering + **stochastic perturbations** → nowcast ensemble
- Parallelized computation of ensemble members with dask.
- Support for standard input/output file formats (e.g. NetCDF and ODIM HDF5) used in meteorological agencies.
- An extensive set of modules for statistical post-processing, verification and visualization of nowcasts.

Module	Description
`io`	reading radar composites and writing nowcast files
`motion`	optical flow methods for motion field computation
`extrapolation`	advection-based extrapolation
`timeseries`	time series methods (e.g. AR models)
`noise`	generation of stochastic noise to perturb precipitation and motion fields
`cascade`	scale-based decomposition of precipitation fields
`nowcasts`	implementation of nowcasting methods
`postprocessing`	statistical post-processing of nowcasts
`verification`	statistical verification of nowcasts and plotting the results
`visualization`	plotting of precipitation and advection fields
`utils`	miscellaneous utility functions (e.g. converting and transforming data and computing the FFT)



Typical workflow for generating nowcasts using pysteps. Boxes with gray colors are applicable to all nowcasting methods. Green colors represent the operations done when scale filtering is used (the S-PROG method, Seed 2003). Blue colors represent the additional operations required to compute stochastic nowcasts (the STEPS model, Bowler et al. 2006). The operations between the input and output are done internally within the pysteps nowcasting methods.

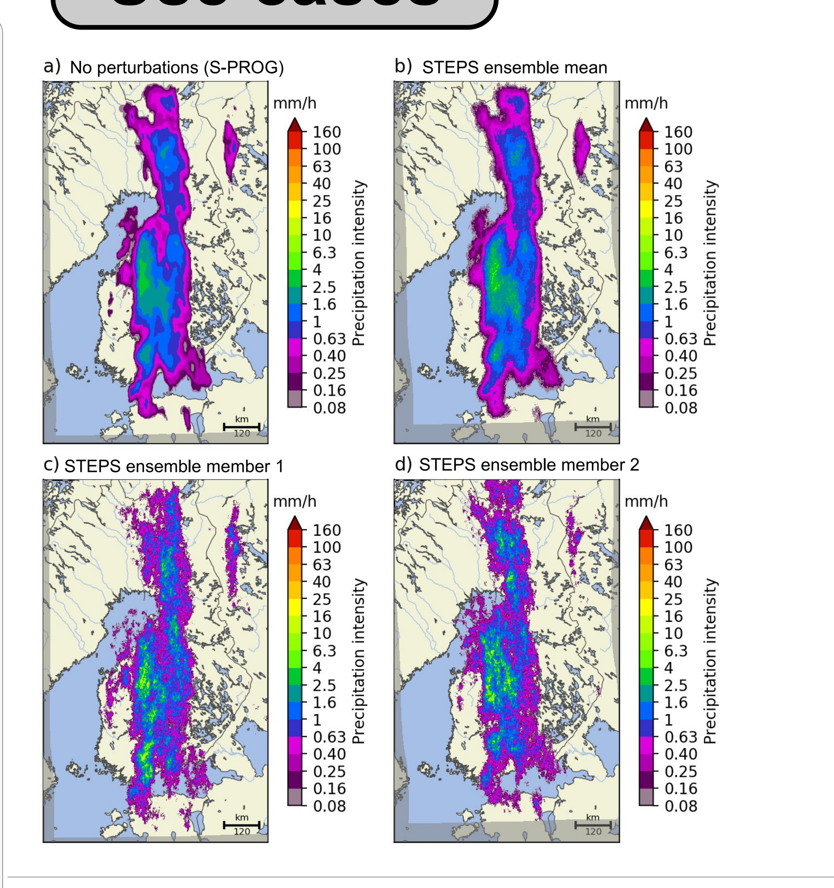
Open-source development model

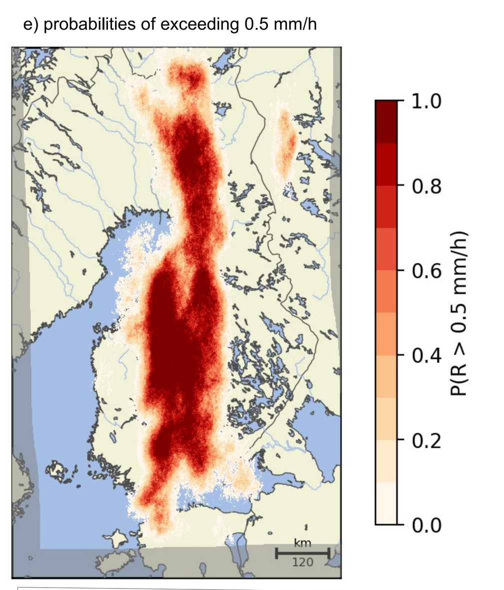
- pysteps is implemented in Python by using numpy, scipy and matplotlib.
- Parallelization is done with dask.
- The recommended method for installing pysteps is to use the Anaconda distribution.
- Published under the BSD-3 license and hosted on GitHub: https://github.com/pySTEPS/pysteps
- Documentation is automatically generated by using Read the Docs: https://pysteps.readthedocs.io
- Travis CI and Codecov are used for continuous integration and testing.

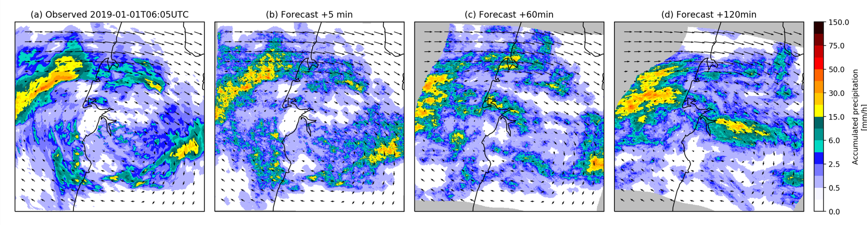
Use cases

The following code example demonstrates the use of pysteps:

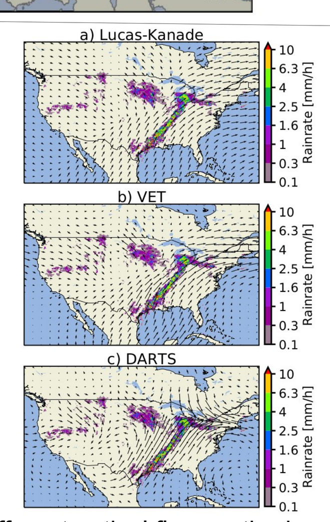
```
from pysteps import io, motion, nowcasts
from pysteps.io import import_fmi_pgm
from pysteps.postprocessing import ensemblestats
from pysteps.utils import conversion, transformation

date = datetime.strptime("201609281600", "%Y%m%d%H%M")
root_path = "pysteps-data/radar/fmi"
fn_pattern = "%Y%m%d%H
%M_fmi.radar.composite.lowest_FIN_SUOMI1"
fn_ext = "pgm.gz"

# find the input files
fns = io.archive.find_by_date(date, root_path,
"%Y%m%d", fn_pattern, fn_ext, 5, num_prev_files=2)

# read the input files, transform units to mm/h and
# take logarithm
Z, _, metadata = io.read_timeseries(fns,
import_fmi_pgm, gzipped=True)
R = conversion.to_rainrate(Z, metadata, 223.0, 1.53)[0]
R = transformation.dB_transform(R, threshold=0.1,
                                zerovalue=-15.0)[0]

# estimate the motion field
oflow_method = motion.get_method("lucaskanade")
V = oflow_method(R)

# compute STEPS nowcast ensemble
nowcast_method = nowcasts.get_method("steps")
R_f = nowcast_method(R, V, 12, n_ens_members=24,
                     R_thr=-10.0, kmperpixel=1.0,
                     timestep=5, num_workers=12)

# transform back to mm/h
R_f = transformation.dB_transform(R_f, threshold=-10.0,
inverse=True)[0]

# compute exceedance probabilities for 0.5 mm/h
P = ensemblestats.excprob(R_f[:, -1, :, :], 0.5)
```





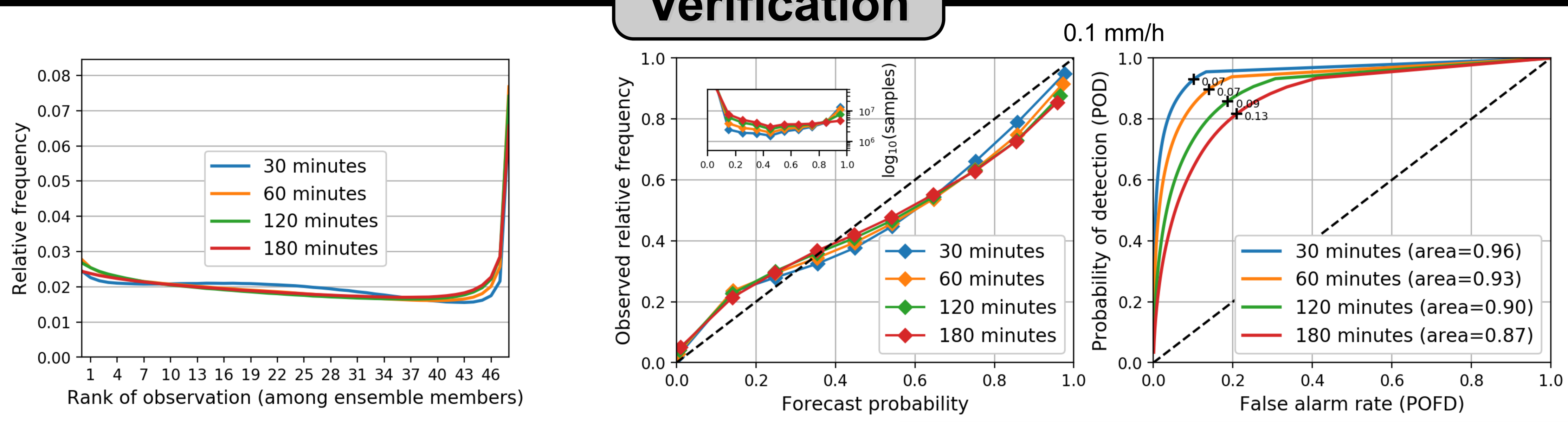


Tropical cyclone event in Australia.

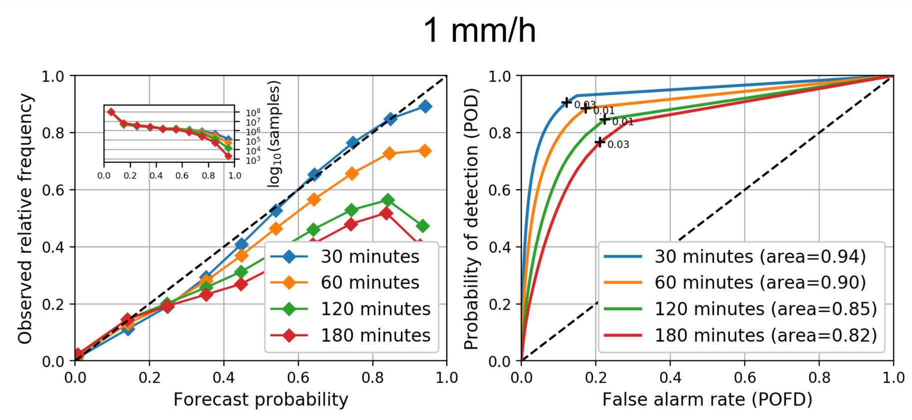


Different optical flow methods applied to US radar composites.

Verification



- pysteps nowcasts have been verified by using the FMI data.
- 24 ensemble members were generated with precipitation intensity thresholds 0.1 mm/h and 1 mm/h.
- For all lead times, less than 15% of the verifying observations fall outside the ensembles.
- For the 0.1 mm/h threshold, the nowcasts are accurate up to 3 hours.
- When the threshold is increased to 1 mm/h, significant loss of reliability and sharpness can be observed with lead times beyond one hour.



Enhancements to earlier STEPS implementations

- **Several optical flow methods** for motion field estimation (Lucas-Kanade, DARTS and Variational Echo Tracking).
- **Localization:** the statistics of precipitation fields are localized in space (Nerini et al. 2017). This gives improved nowcast accuracy in larger domains with different types of precipitation (e.g. stratiform and convective).
- **Computational improvements:** the cascade decomposition, AR(2) models and noise generation can be applied in the spectral domain to improve performance and reduce memory usage (Pulkkinen et al. 2019).

Publications

- S. Pulkkinen, D. Nerini, A. Pérez Hortal, C. Velasco-Forero, A. Seed, U. Germann, L. Foresti, pysteps: an open-source Python library for probabilistic precipitation nowcasting (v1.0), submitted to Geoscientific Model Development, 2019.
- D. Nerini, N. Besic, I. Sideris, U. Germann, L. Foresti, A non-stationary stochastic ensemble generator for radar rainfall fields based on the short-space Fourier transform, Hydrology and Earth System Sciences, 21(6), 2777-2797, 2017.
- S. Pulkkinen, V. Chandrasekar, A.-M. Harri, Stochastic Spectral Method for Radar-Based Probabilistic Precipitation Nowcasting, Journal of Atmospheric and Oceanic Technology, Early Online Release, doi:10.1175/JTECH-D-18-0242.1, 2019.