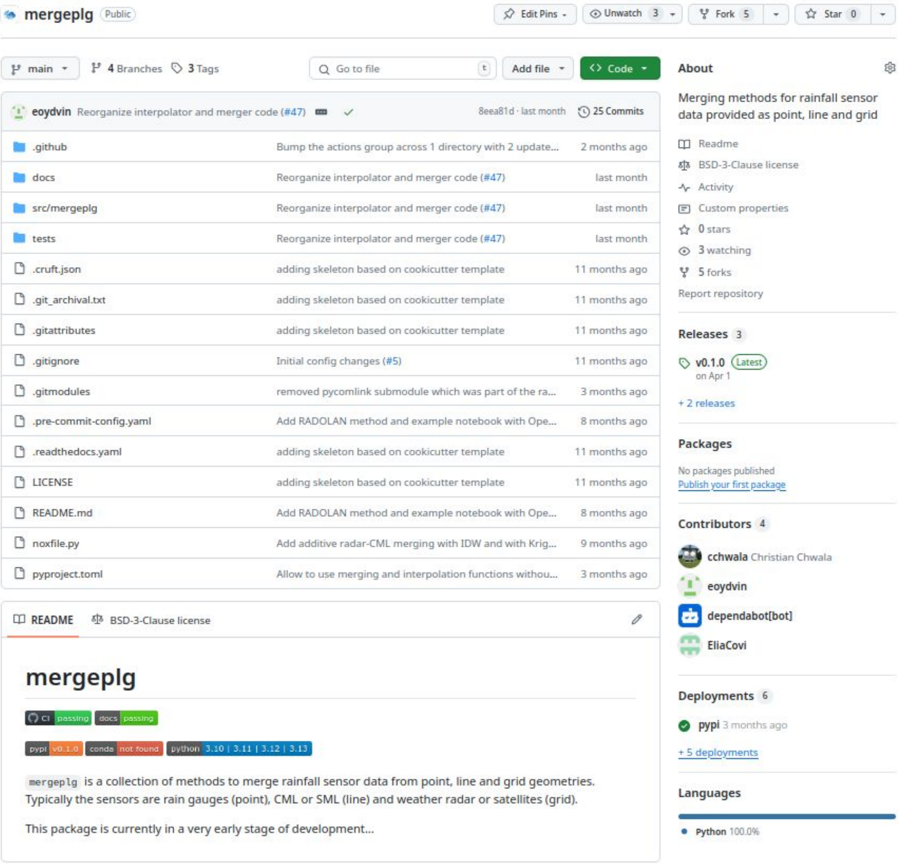
Merging weather radar fields with data from commercial microwave links using mergeplg

Erlend Øydvin, Maximilian Graf, Christian Chwala, Elia Covi

mergeplg

- A OpenSense python package for
  - Interpolation
    - Ground observations
  - Merging
    - Radar + ground observations
- Lives in the OpenSense ecosystem
- Comparison study



eoydvin Reorganize interpolator and merger code (#47) 8eea81d · last month 25 Commits

.github	Bump the actions group across 1 directory with 2 update...	2 months ago
docs	Reorganize interpolator and merger code (#47)	last month
src/mergeplg	Reorganize interpolator and merger code (#47)	last month
tests	Reorganize interpolator and merger code (#47)	last month
.cruft.json	adding skeleton based on cookicutter template	11 months ago
.git_archival.txt	adding skeleton based on cookicutter template	11 months ago
.gitattributes	adding skeleton based on cookicutter template	11 months ago
.gitignore	Initial config changes (#5)	11 months ago
.gitmodules	removed pycomlink submodule which was part of the ra...	3 months ago
.pre-commit-config.yaml	Add RADOLAN method and example notebook with Ope...	8 months ago
.readthedocs.yaml	adding skeleton based on cookicutter template	11 months ago
LICENSE	adding skeleton based on cookicutter template	11 months ago
README.md	Add RADOLAN method and example notebook with Ope...	8 months ago
noxfile.py	Add additive radar-CML merging with IDW and with Krig...	9 months ago
pyproject.toml	Allow to use merging and interpolation functions withou...	3 months ago

mergeplg

CI passing docs passing

pypi v0.1.0 conda not found python 3.10 | 3.11 | 3.12 | 3.13

`mergeplg` is a collection of methods to merge rainfall sensor data from point, line and grid geometries. Typically the sensors are rain gauges (point), CML or SML (line) and weather radar or satellites (grid).

This package is currently in a very early stage of development...

About

Merging methods for rainfall sensor data provided as point, line and grid

Readme
BSD-3-Clause license
Activity
Custom properties
0 stars
3 watching
5 forks
Report repository

Releases 3

v0.1.0 Latest
on Apr 1

+ 2 releases

Packages

No packages published
Publish your first package

Contributors 4

cchwala Christian Chwala
eoydvin
dependabot[bot]
EliaCovi

Deployments 6

pypi 3 months ago

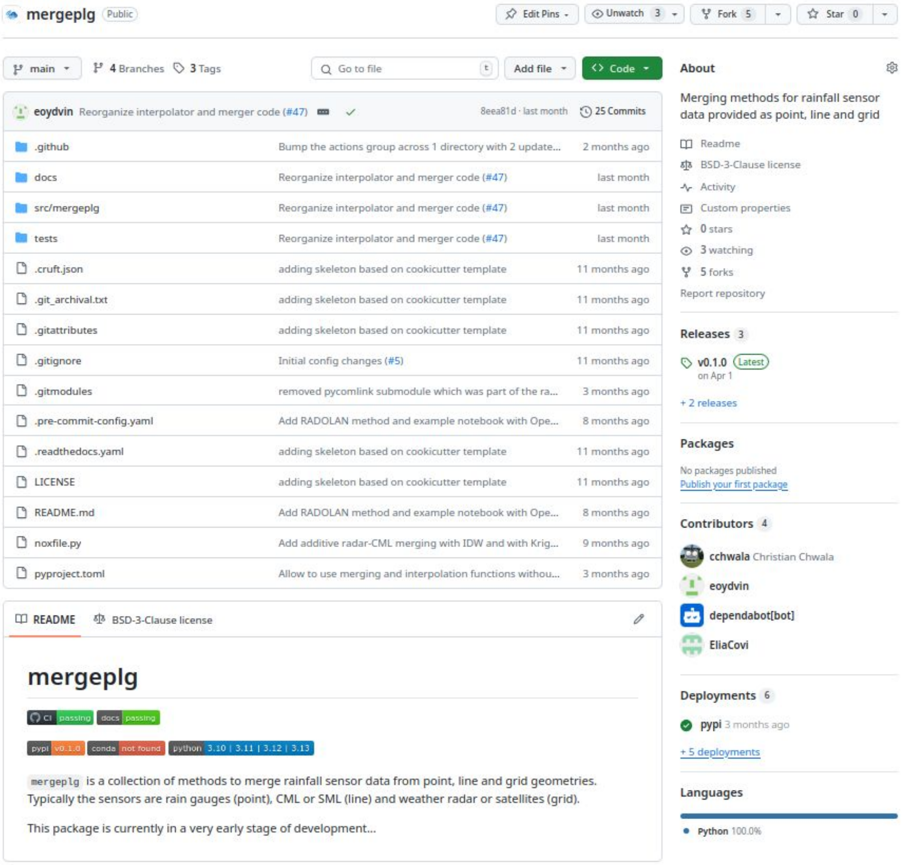
+ 5 deployments

Languages

Python 100.0%

mergeplg

- A OpenSense python package for
  - Interpolation
    - Ground observations
  - Merging
    - Radar + ground observations
- Lives in the OpenSense ecosystem
- Comparison study



eoydvin Reorganize interpolator and merger code (#47) ✓ 8eea81d · last month 25 Commits

.github	Bump the actions group across 1 directory with 2 update...	2 months ago
docs	Reorganize interpolator and merger code (#47)	last month
src/mergeplg	Reorganize interpolator and merger code (#47)	last month
tests	Reorganize interpolator and merger code (#47)	last month
.cruft.json	adding skeleton based on cookicutter template	11 months ago
.git_archival.txt	adding skeleton based on cookicutter template	11 months ago
.gitattributes	adding skeleton based on cookicutter template	11 months ago
.gitignore	Initial config changes (#5)	11 months ago
.gitmodules	removed pycomlink submodule which was part of the ra...	3 months ago
.pre-commit-config.yaml	Add RADOLAN method and example notebook with Ope...	8 months ago
.readthedocs.yaml	adding skeleton based on cookicutter template	11 months ago
LICENSE	adding skeleton based on cookicutter template	11 months ago
README.md	Add RADOLAN method and example notebook with Ope...	8 months ago
noxfile.py	Add additive radar-CML merging with IDW and with Krig...	9 months ago
pyproject.toml	Allow to use merging and interpolation functions withou...	3 months ago

mergeplg

CI passing | docs passing

pypi v0.1.0 | conda not found | python 3.10 | 3.11 | 3.12 | 3.13

`mergeplg` is a collection of methods to merge rainfall sensor data from point, line and grid geometries. Typically the sensors are rain gauges (point), CML or SML (line) and weather radar or satellites (grid).

This package is currently in a very early stage of development...

Merging methods for rainfall sensor data provided as point, line and grid

0 stars

3 watching

5 forks

Releases 3

v0.1.0 Latest
on Apr 1

+ 2 releases

Packages

No packages published

Contributors 4

cchwala Christian Chwala
eoydvin
dependabot[bot]
EliaCovi

Deployments 6

✓ pypi 3 months ago

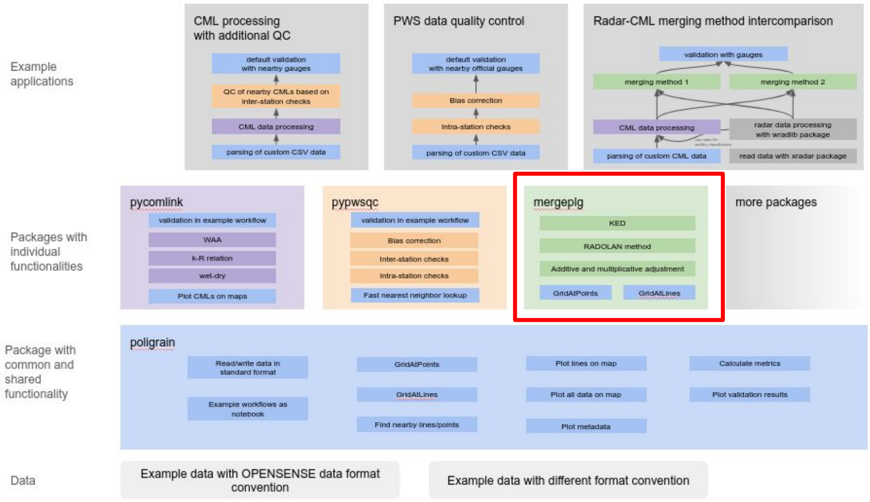
+ 5 deployments

Languages

● Python 100.0%

mergeplg

- A OpenSense python package for
    - Interpolation
        - Ground observations
    - Merging
        - Radar + ground observations
- Lives in the OpenSense ecosystem
- Comparison study



Example applications

CML processing with additional QC

default validation with nearby gauges

QC of nearby CMLs based on inter-station checks

CML data processing

parsing of custom CSV data

PWS data quality control

default validation with nearby official gauges

Bias correction

Intra-station checks

parsing of custom CSV data

Radar-CML merging method intercomparison

validation with gauges

merging method 1

merging method 2

CML data processing

radar data processing with wradlib package

parsing of custom CML data

read data with xradar package

Packages with individual functionalities

pycomlink

validation in example workflow

WAA

k-R relation

wet-dry

Plot CMLs on maps

pypwsqc

validation in example workflow

Bias correction

Inter-station checks

Intra-station checks

Fast nearest neighbor lookup

mergeplg

KED

RADOLAN method

Additive and multiplicative adjustment

GridAtPoints

GridAtLines

more packages

Package with common and shared functionality

poligrain

Read/write data in standard format

Example workflows as notebook

GridAtPoints

GridAtLines

Find nearby lines/points

Plot lines on map

Plot all data on map

Plot metadata

Calculate metrics

Plot validation results

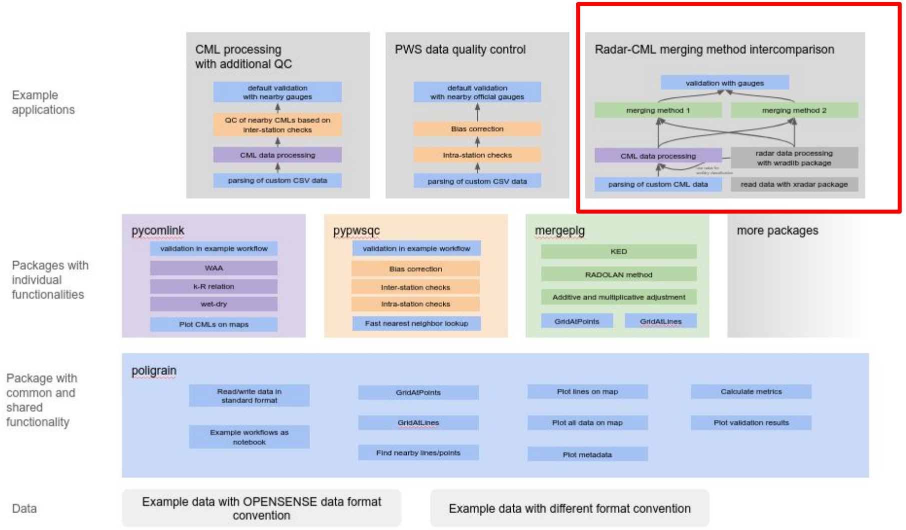
Data

Example data with OPENSENSE data format convention

Example data with different format convention

mergeplg

- A OpenSense python package for
    - Interpolation
        - Ground observations
    - Merging
        - Radar + ground observations
- Lives in the OpenSense ecosystem
- Comparison study


Example applications
CML processing with additional QC
default validation with nearby gauges
QC of nearby CMLs based on inter-station checks
CML data processing
parsing of custom CSV data
PWS data quality control
default validation with nearby official gauges
Bias correction
Intra-station checks
parsing of custom CSV data
Radar-CML merging method intercomparison
validation with gauges
merging method 1
merging method 2
CML data processing
radar data processing with wradlib package
parsing of custom CML data
read data with xradar package
Packages with individual functionalities
pycomlink
validation in example workflow
WAA
k-R relation
wet-dry
Plot CMLs on maps
pypwsqc
validation in example workflow
Bias correction
Inter-station checks
Intra-station checks
Fast nearest neighbor lookup
mergeplg
KED
RADOLAN method
Additive and multiplicative adjustment
GridAtPoints
GridAtLines
more packages
Package with common and shared functionality
poligrain
Read/write data in standard format
Example workflows as notebook
GridAtPoints
GridAtLines
Find nearby lines/points
Plot lines on map
Plot all data on map
Plot metadata
Calculate metrics
Plot validation results
Data
Example data with OPENSENSE data format convention
Example data with different format convention

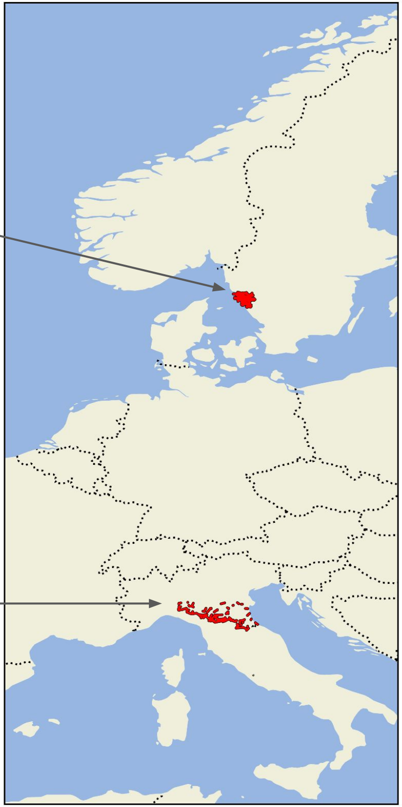
Data

OpenMRG

Adersson et al. (2022)

OpenRainER

Covi et al. (2025)

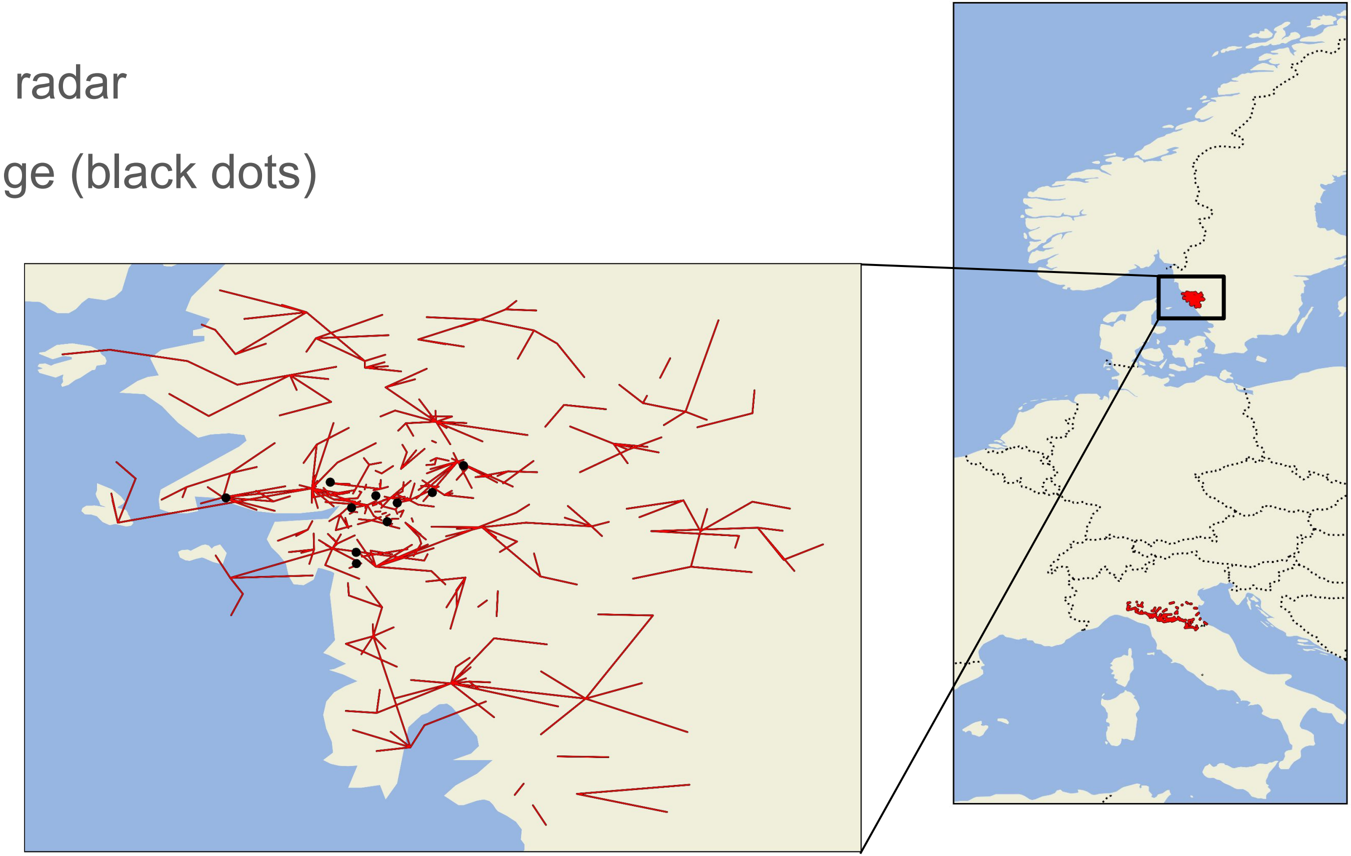


Data - OpenMRG

350 CMLs (red lines) + radar

Reference: 10 rain gauge (black dots)

August 2015

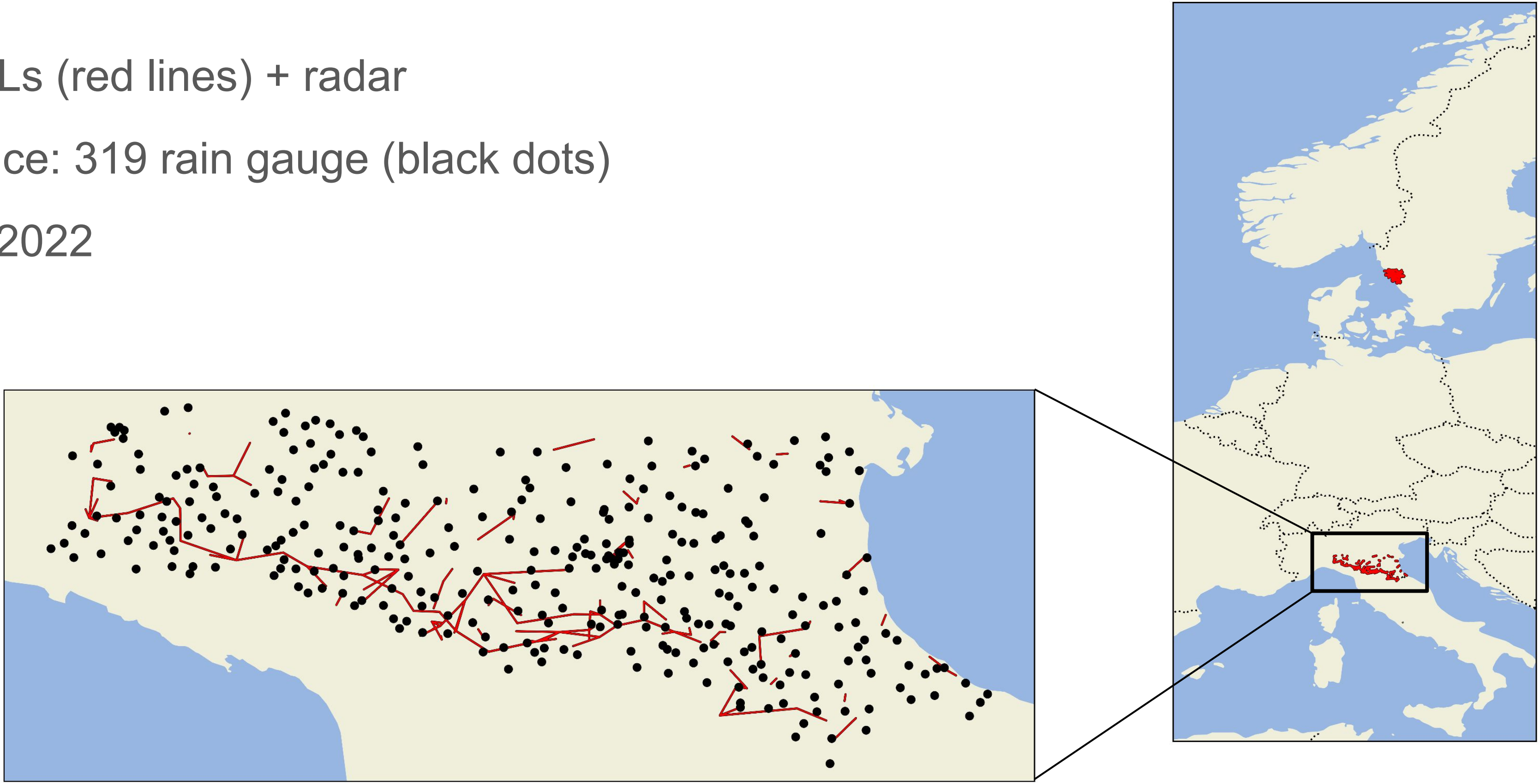


Data - OpenRainER

114 CMLs (red lines) + radar

Reference: 319 rain gauge (black dots)

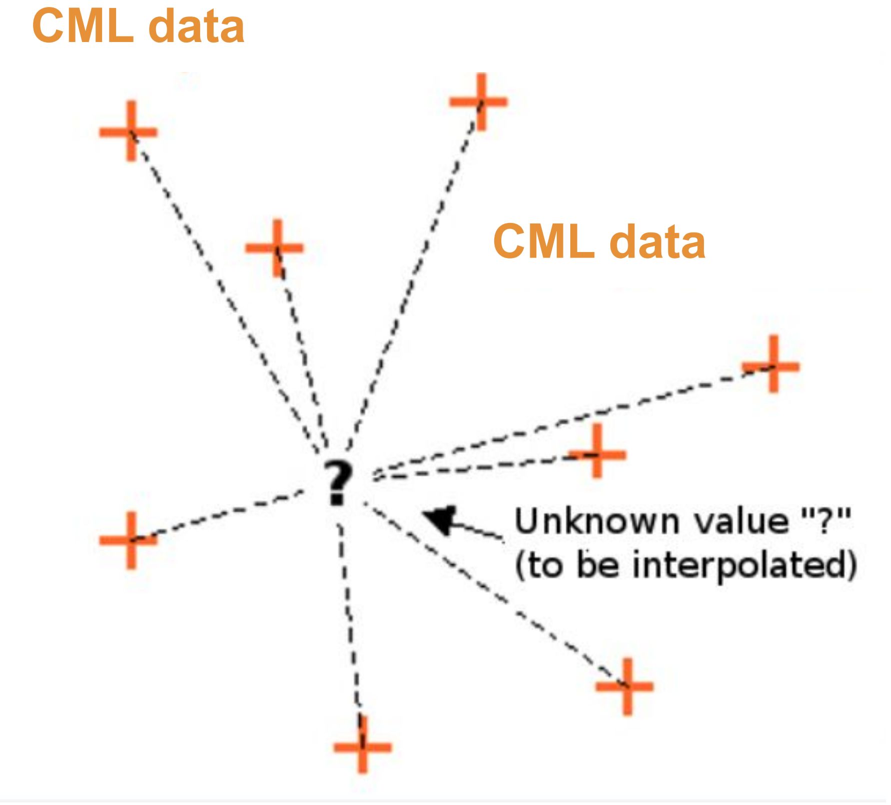
August 2022



Interpolation methods

Goal: Optimally weight CML data to estimate unknown value

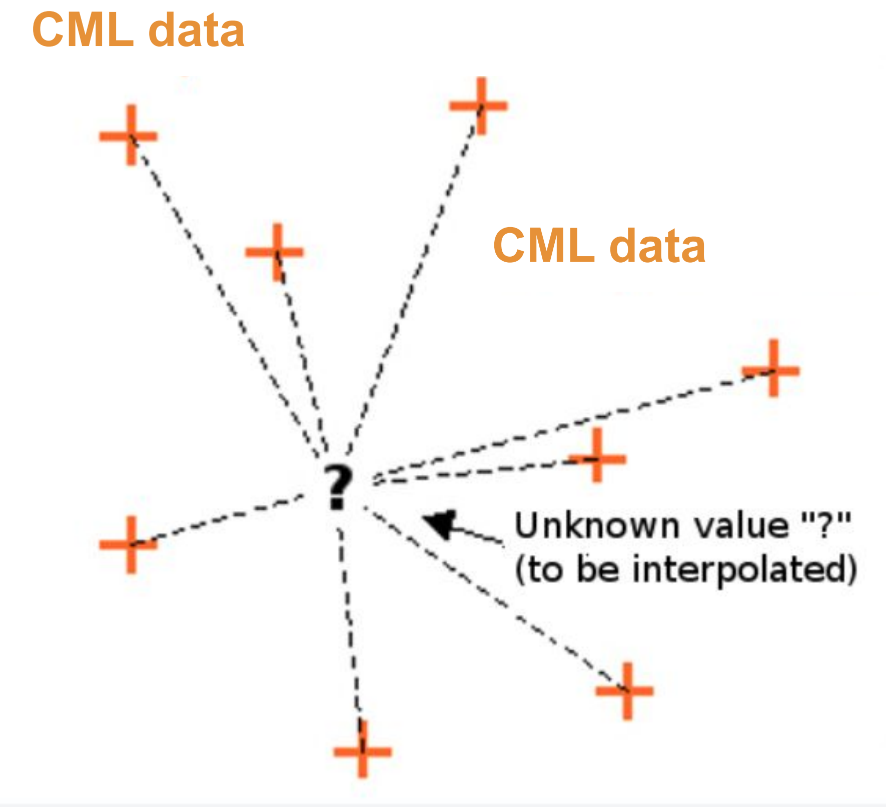
1. interpolateIDW
   a. Uses inverse distance weighting
   b. CML midpoint
2. interpolateBKpoint
   a. Uses Ordinary Kriging
   b. CML midpoints
3. interpolateBKline
   a. Uses Ordinary Block Kriging
   b. CML line



Interpolation methods

Goal: Optimally weight CML data to estimate unknown value

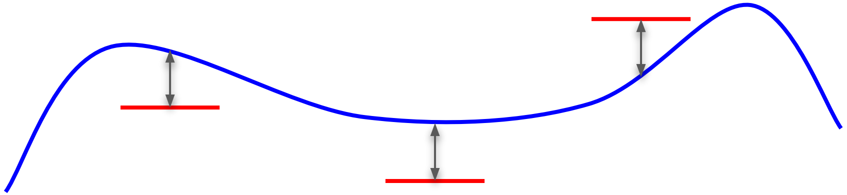
1. interpolateIDW
    a. Uses inverse distance weighting
    b. CML midpoint
2. interpolateBKpoint
    a. Uses Ordinary Kriging
    b. CML midpoints
3. interpolateBKline
    a. Uses Ordinary Block Kriging
    b. CML line



Merging methods

Additive merging:

- Estimate difference between radar and CML
- Interpolate difference (IDW, Kriging)
- Add interpolated difference to radar map

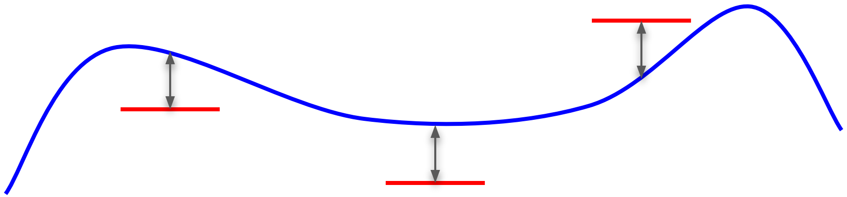


1. mergeIDW
   a. Uses inverse distance weighting
   b. CML midpoint
2. mergeBKpoint
   a. Uses Ordinary Kriging
   b. CML midpoints
3. mergeBKline
   a. Uses Ordinary Block Kriging
   b. CML line

Merging methods

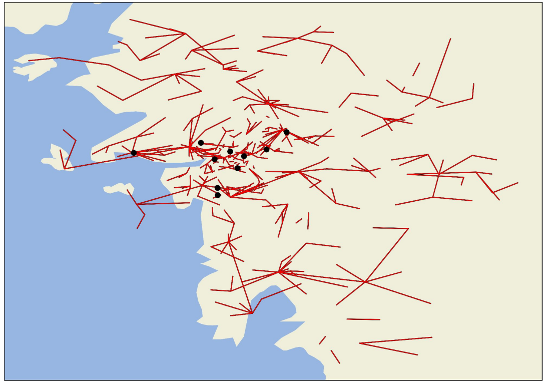
Additive merging:

- Estimate difference between radar and CML
- Interpolate difference (IDW, Kriging)
- Add interpolated difference to radar map



1. mergeIDW
    1. Uses inverse distance weighting
    2. CML midpoint
2. mergeBKpoint
    1. Uses Ordinary Kriging
    2. CML midpoints
3. mergeBKline
    1. Uses Ordinary Block Kriging
    2. CML line

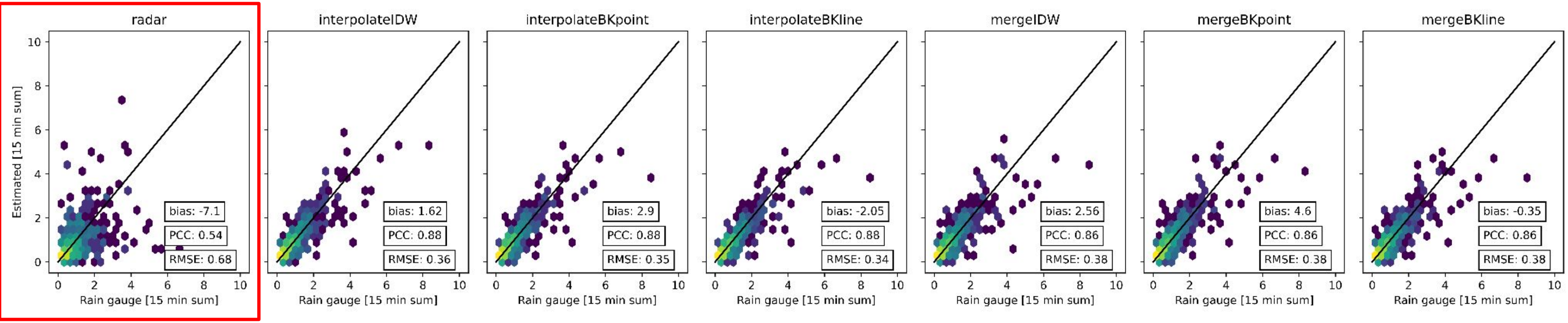
Results OpenMRG



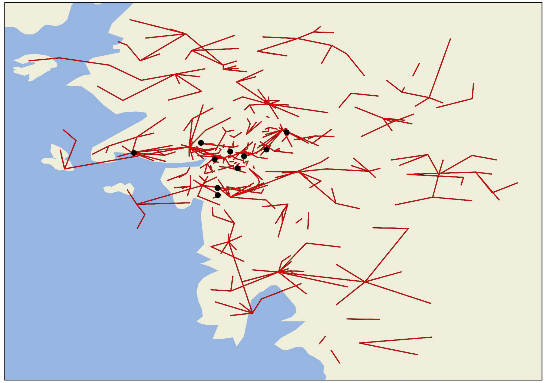
Radar performs good

Interpolated CMLs performs better than radar

Radar merged with CMLs performs similarly good as interpolated CML data



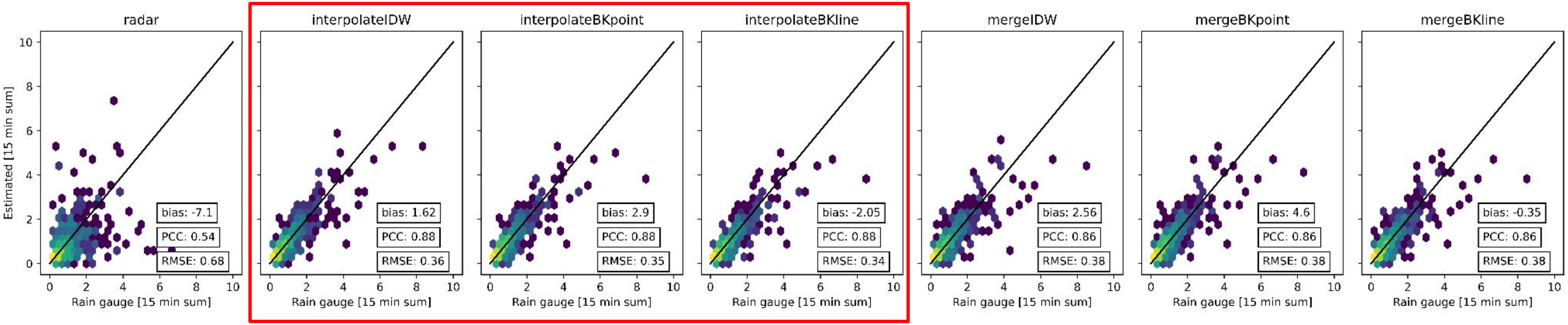
Results OpenMRG



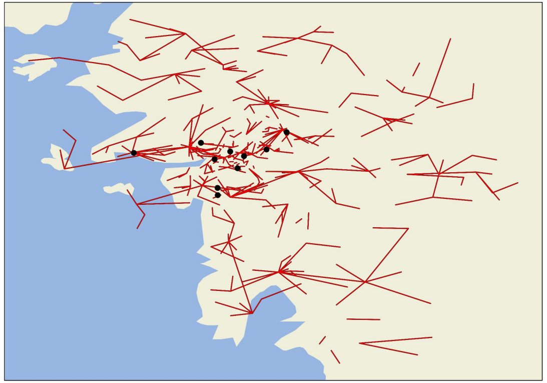
Radar performs good

Interpolated CMLs performs better than radar

Radar merged with CMLs performs similarly good as interpolated CML data



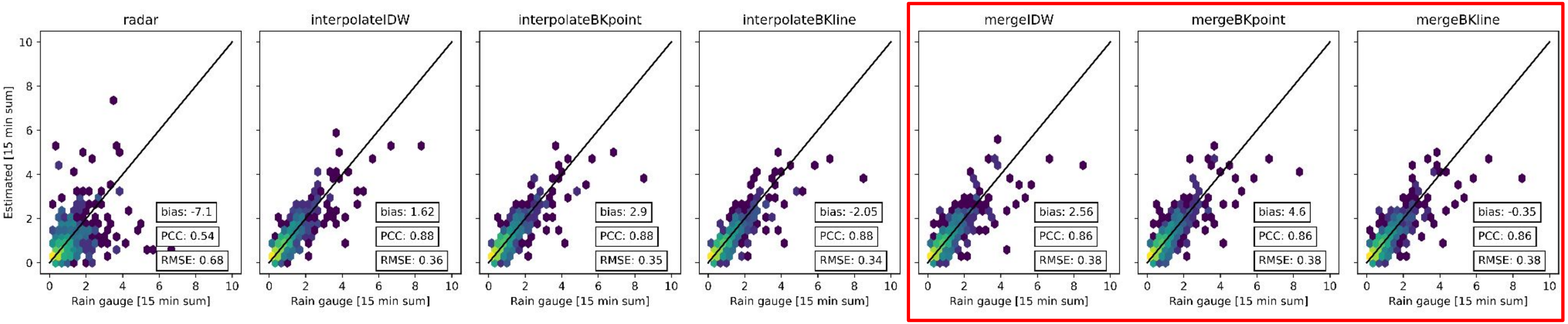
Results OpenMRG



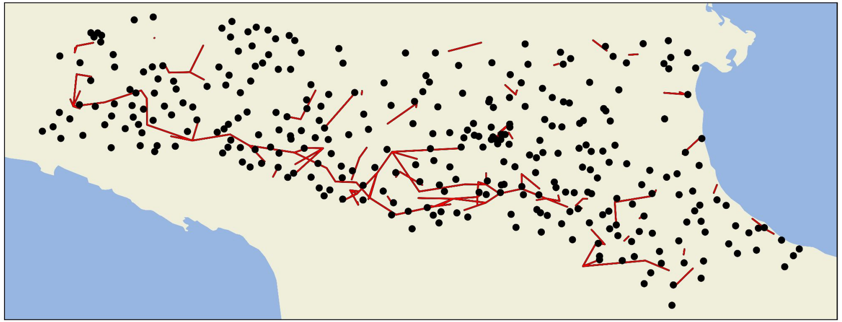
Radar performs good

Interpolated CMLs performs better than radar

Radar merged with CMLs performs similarly good as interpolated CML data



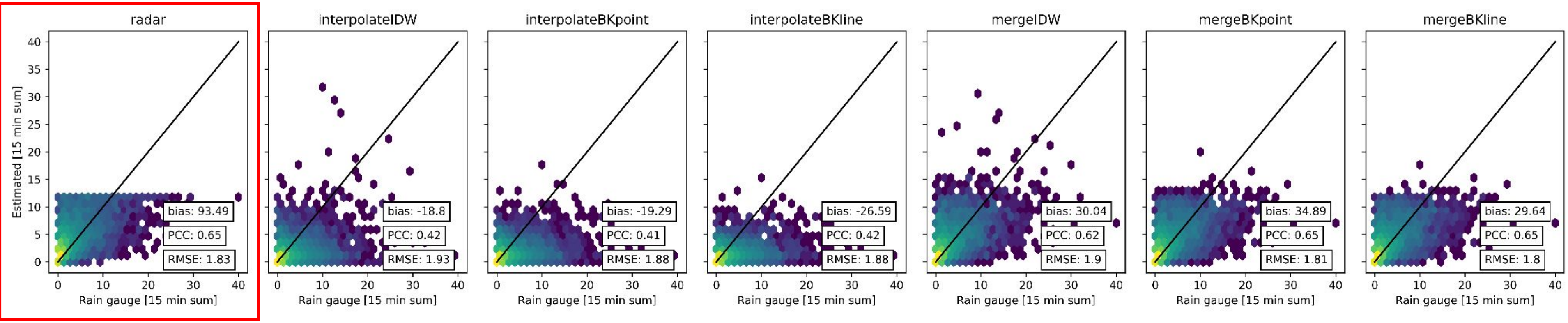
Results OpenRainER



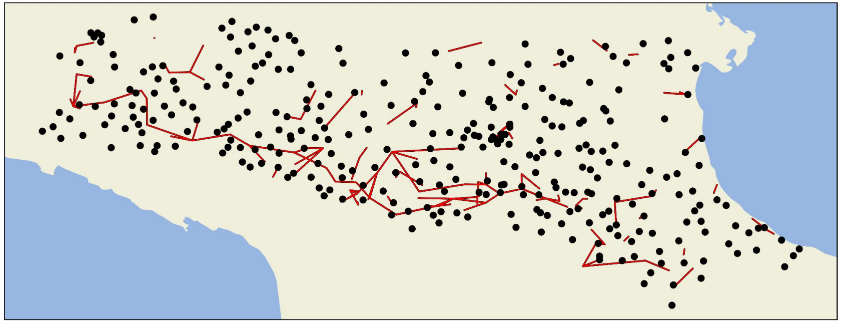
Higher rainfall amounts

Interpolated CMLs, less good than radar

Radar merged with CMLs, better bias than radar



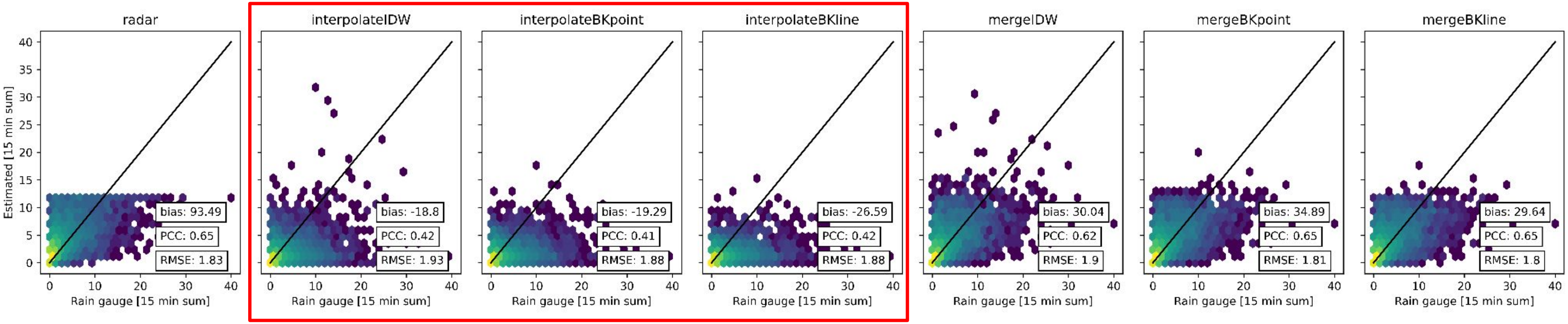
Results OpenRainER



Higher rainfall amounts

Interpolated CMLs, less good than radar

Radar merged with CMLs, better bias than radar

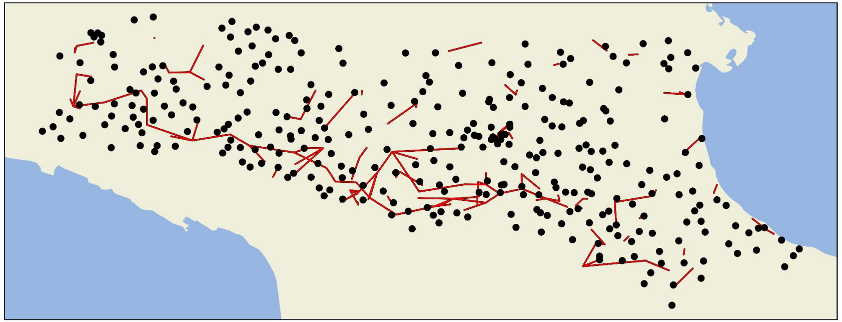


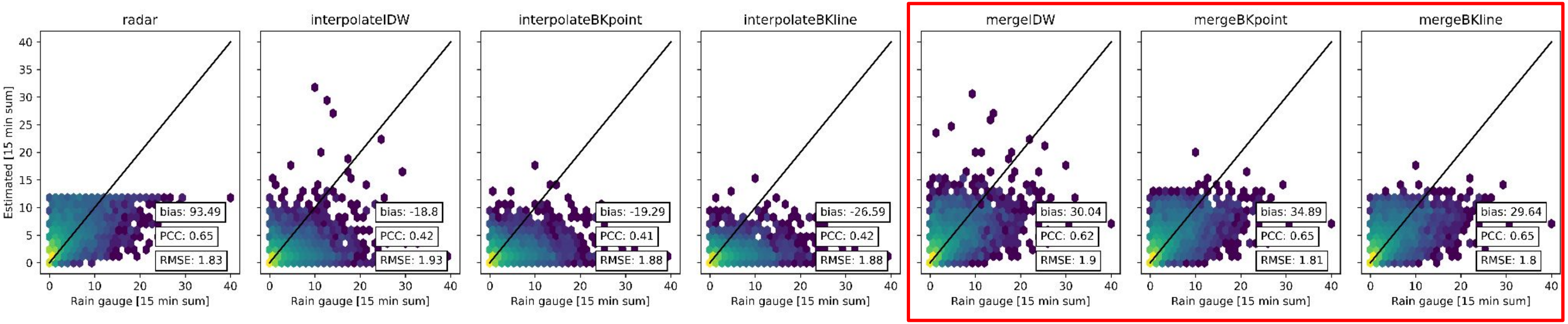
Results OpenRainER

Higher rainfall amounts

Interpolated CMLs, less good than radar

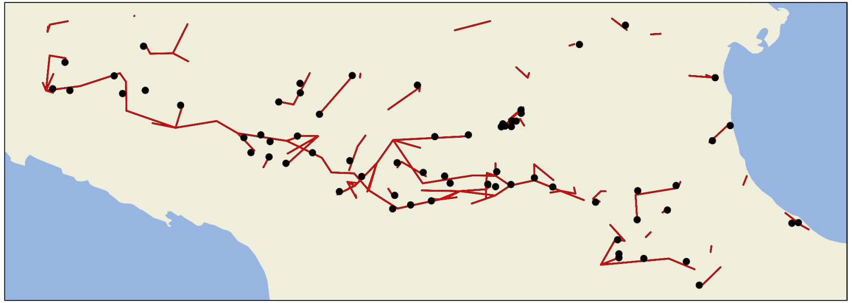
Radar merged with CMLs, better bias than radar





Results OpenRainER

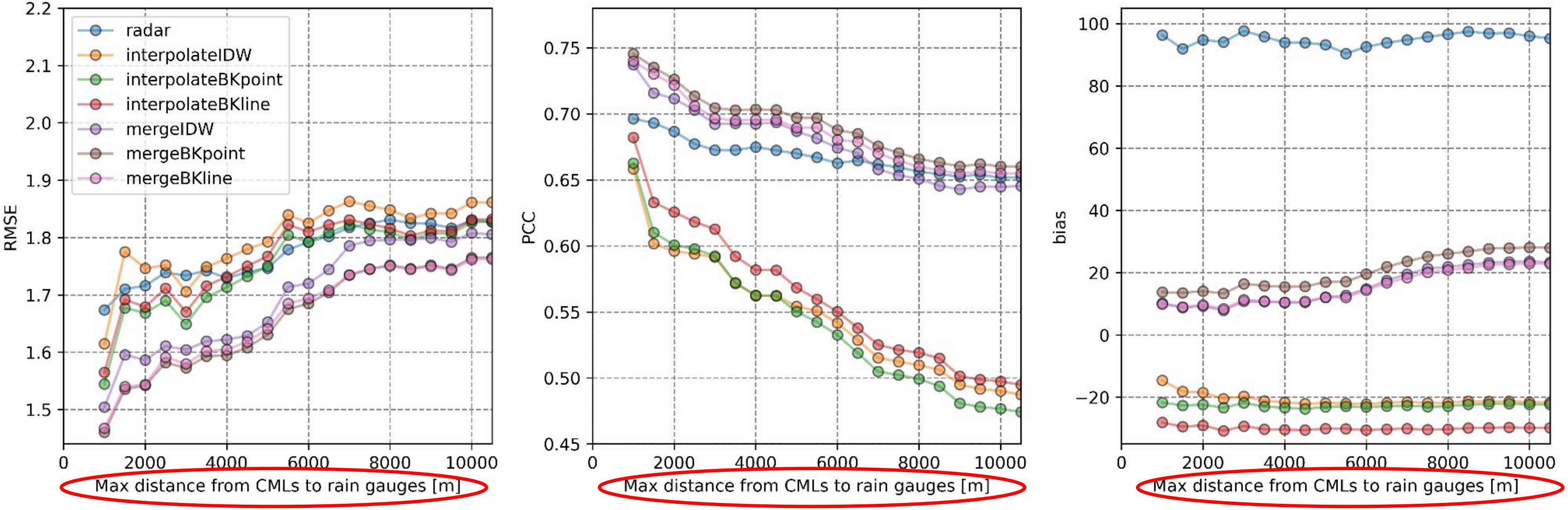
Use only rain gauges within 2 km of a CML



Analysis OpenRainER

Rain gauges within **x** meters of a CML

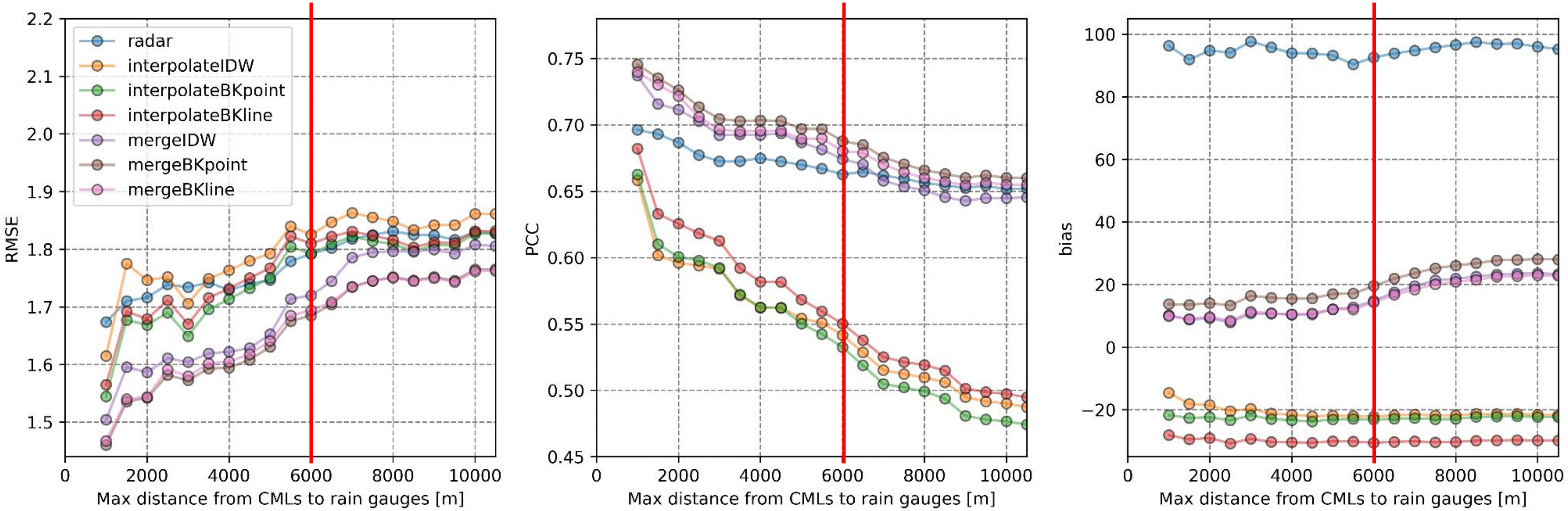
- For instance at 6000 m, only rain gauges within 6000 m of a CML is included
- Evaluated RMSE, PCC and bias



Analysis OpenRainER

Rain gauges within **x** meters of a CML

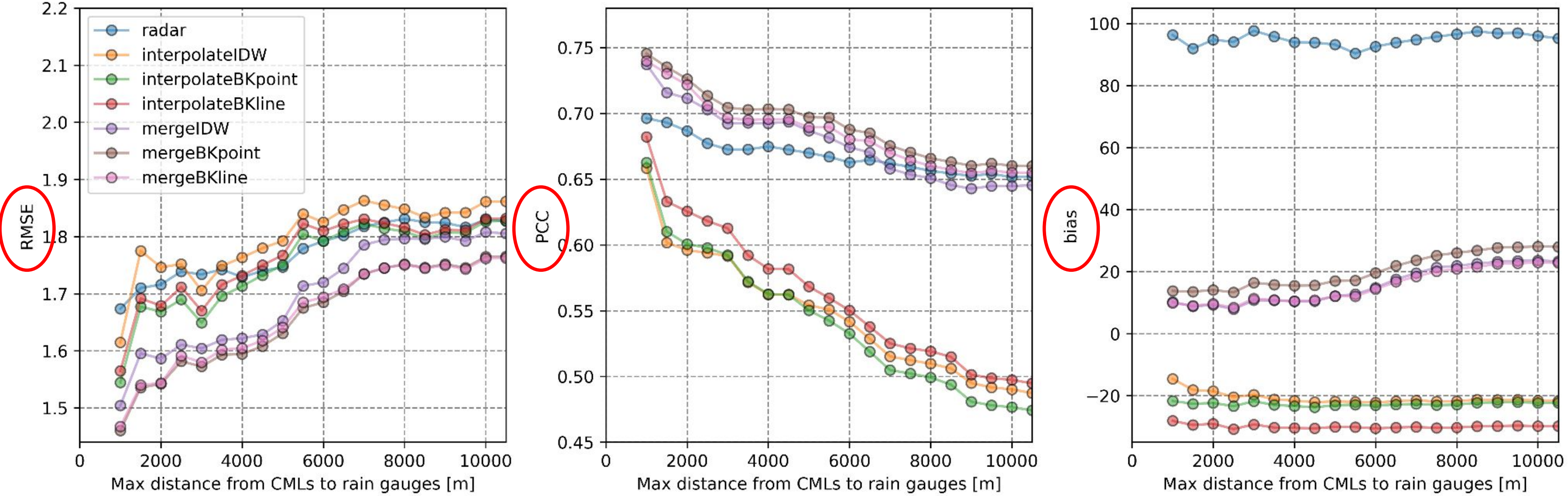
- For instance at 6000 m, only rain gauges within 6000 m of a CML is included
- Evaluated RMSE, PCC and bias



Analysis OpenRainER

Rain gauges within **x** meters of a CML

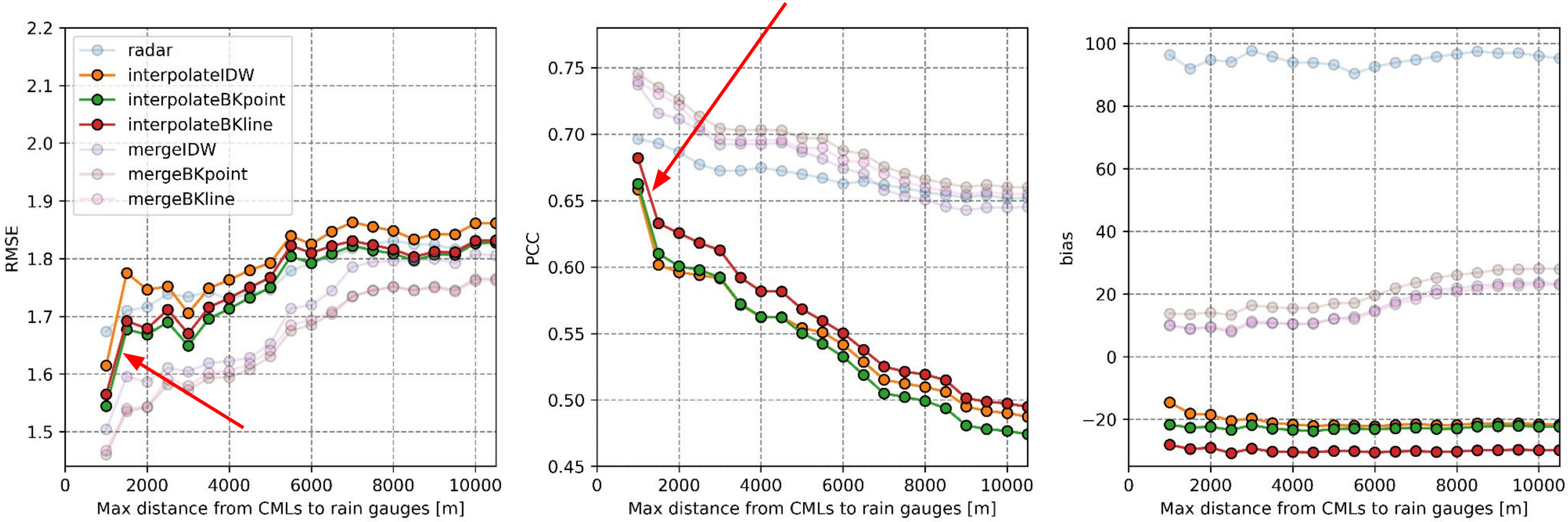
- For instance at 6000 m, only rain gauges within 6000 m of a CML is included
- Evaluated RMSE, PCC and bias



Analysis OpenRainER

Interpolated CML data

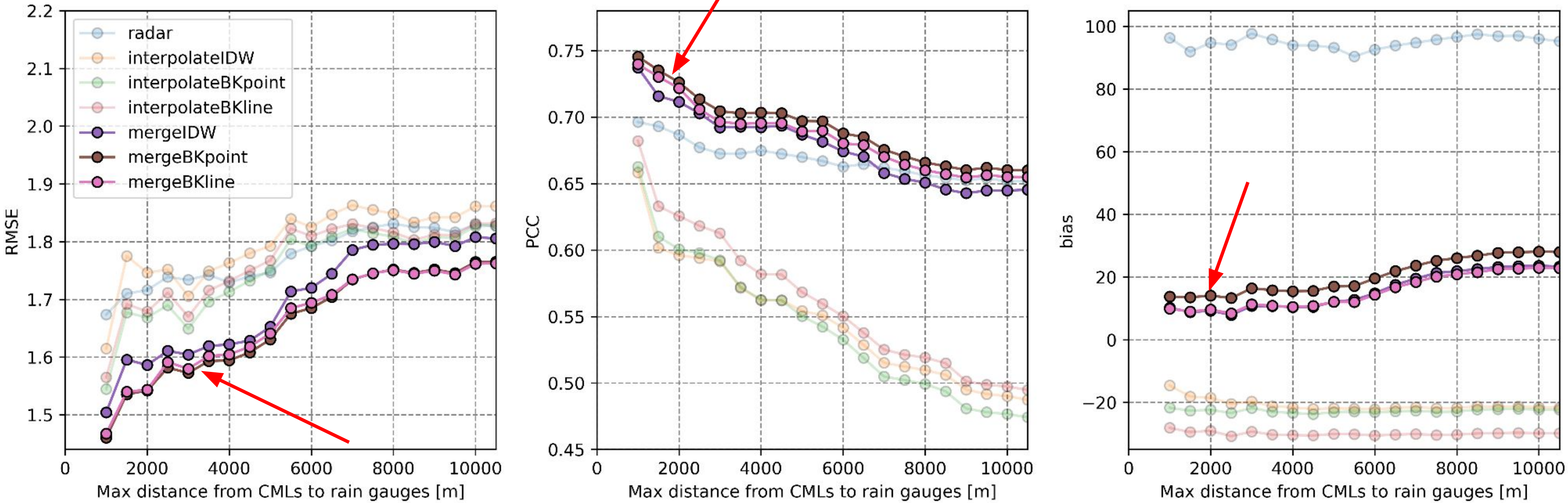
- Improves as distance decreases
- Kriging slightly better



Analysis OpenRainER

Merged CML and radar data

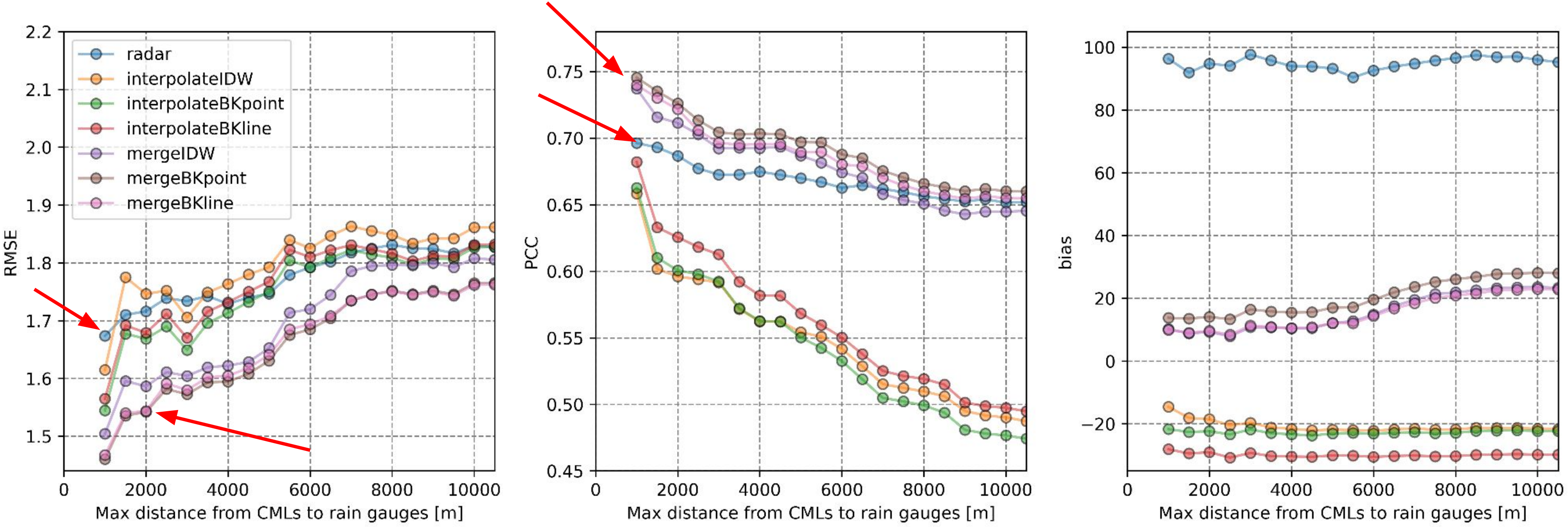
- Also improves as distance decreases
- Kriging slightly better



Analysis OpenRainER

When distance **decreases**

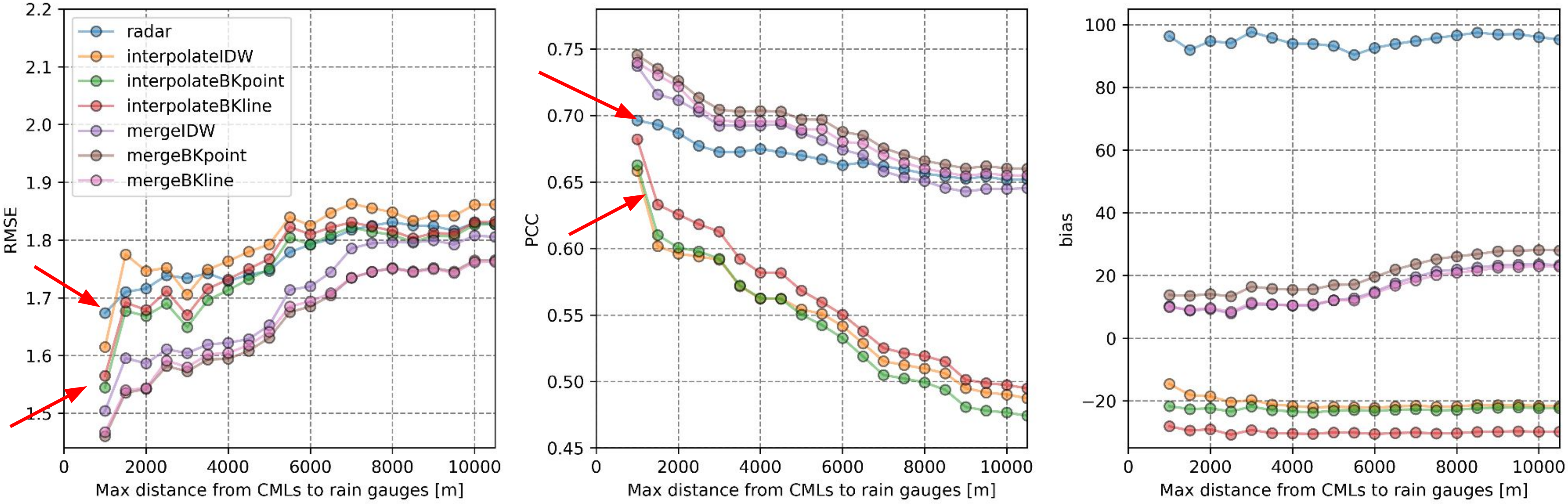
- Merged fields outperforms radar
- Interpolated fields almost as good as radar



Analysis OpenRainER

When distance **decreases**

- Merged fields outperforms radar
- Interpolated fields almost as good as radar



Conclusions and perspectives

- CMLs perform good when there are many CMLs close to the reference
  - OpenMRG (radar PCC: 0.54, interpolateIDW PCC: 0.88)
- Merging radar with CML improves rainfall fields
  - OpenRainER (radar PCC = 0.7, mergeBKpoint approaching 0.75)
- OpenRainER seem to approach OpenMRG results
  - Further work: Investigate how CML densities affects the results
- Kriging variants slightly better than IDW
  - Event based analysis to better understand the strength and weaknesses

Conclusions and perspectives

- CMLs perform good when there are many CMLs close to the reference
  - OpenMRG (radar PCC: 0.54, interpolateIDW PCC: 0.88)
- Merging radar with CML improves rainfall fields
  - OpenRainER (radar PCC = 0.7, mergeBKpoint approaching 0.75)
- OpenRainER seem to approach OpenMRG results
  - Further work: Investigate how CML densities affects the results
- Kriging variants slightly better than IDW
  - Event based analysis to better understand the strength and weaknesses

Conclusions and perspectives

- CMLs perform good when there are many CMLs close to the reference
    - OpenMRG (radar PCC: 0.54, interpolateIDW PCC: 0.88)
- Merging radar with CML improves rainfall fields
    - OpenRainER (radar PCC = 0.7, mergeBKpoint approaching 0.75)
- OpenRainER seem to approach OpenMRG results
    - Further work: Investigate how CML densities affects the results
- Kriging variants slightly better than IDW
    - Event based analysis to better understand the strength and weaknesses

Conclusions and perspectives

- CMLs perform good when there are many CMLs close to the reference
    - OpenMRG (radar PCC: 0.54, interpolateIDW PCC: 0.88)
- Merging radar with CML improves rainfall fields
    - OpenRainER (radar PCC = 0.7, mergeBKpoint approaching 0.75)
- OpenRainER seem to approach OpenMRG results
    - Further work: Investigate how CML densities affects the results
- Kriging variants slightly better than IDW
    - Event based analysis to better understand the strength and weaknesses

Thank you for your attention!