ML4AQ
(Machine Learning for Air Quality)

Herve Petetin - Octobre 2018

Goal : Explore the use of ML for ***forecasting*** and ideally ***better understanding*** the errors of AQ models

Application to MONARCH

Reference bias forecasting system : Kalman Filter (KF), currently used in the operational CALIOPE system (black box…)

➔ Pre-requisite : Need to develop a stand-alone KF version consistent with the one currently used in CALIOPE

What has been done?

➔ New stand-alone version of Kalman Filter (hereafter called ***modkf1***) coded in R (detailed notice in progress)

➔ Comparison with operational CALIOPE-KF (hereafter called ***modkf0***) time series

NB1 : CALIOPE data available only since February 2018

NB2 : Scripts are parallelized on power9, thus easy and fast to analyse large amount of stations

A few words on Kalman filter

General formulation of the problem :

$$\begin{cases} \mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \boldsymbol{\eta}_t & (1) \\ \mathbf{y}_t = \mathbf{H}_t \mathbf{x}_{t-1} + \boldsymbol{\epsilon}_t & (2) \end{cases}$$

with $\mathbf{x}_t$ the systematic error between observations and forecasts (unknown, not observable), $\mathbf{F}_t$ the system matrix, $\boldsymbol{\eta}_t$ the random change from time $(t-1)$ to time t, $\mathbf{y}_t$ the observation of the error between observation and forecast, $\mathbf{H}_t$ the observation matrix, $\boldsymbol{\epsilon}_t$ the random observation error. Both $\boldsymbol{\eta}_t$ and $\boldsymbol{\epsilon}_t$ are considered independent, time-independent and correspond to a white Gaussian noise drawn from zero-mean normal distributions associated with the covariance matrices $\mathbf{W}_t$ and $\mathbf{V}_t$, respectively (i.e. mathematically : $\boldsymbol{\eta}_t \sim N(0, \mathbf{W}_t)$ and $\boldsymbol{\epsilon}_t \sim N(0, \mathbf{V}_t)$).

[…] Final form of the KF updating equations :

$$\begin{cases} \hat{x}_{t|t} = \hat{x}_{t-1|t-1} + k_t(y_t - \hat{x}_{t-1|t-1}) & (15) \\ \hat{p}_{t|t} = (1-k_t)(\hat{p}_{t-1|t-1} + w_t) & (16) \\ k_t = (\hat{p}_{t-1|t-1} + w_t)(\hat{p}_{t-1|t-1} + w_t + v_t) & (17) \end{cases}$$

The way w_t/v_t is estimated in the KF is crucial!
Many possible approaches exist to estimate this ratio.
Here : offline version : test KF on many w_t/v_t values (e.g. from 0.001 to 100) and selection of the one that minimizes the RMSE or PCC (Pearson correlation coefficient)

A few words on Kalman filter

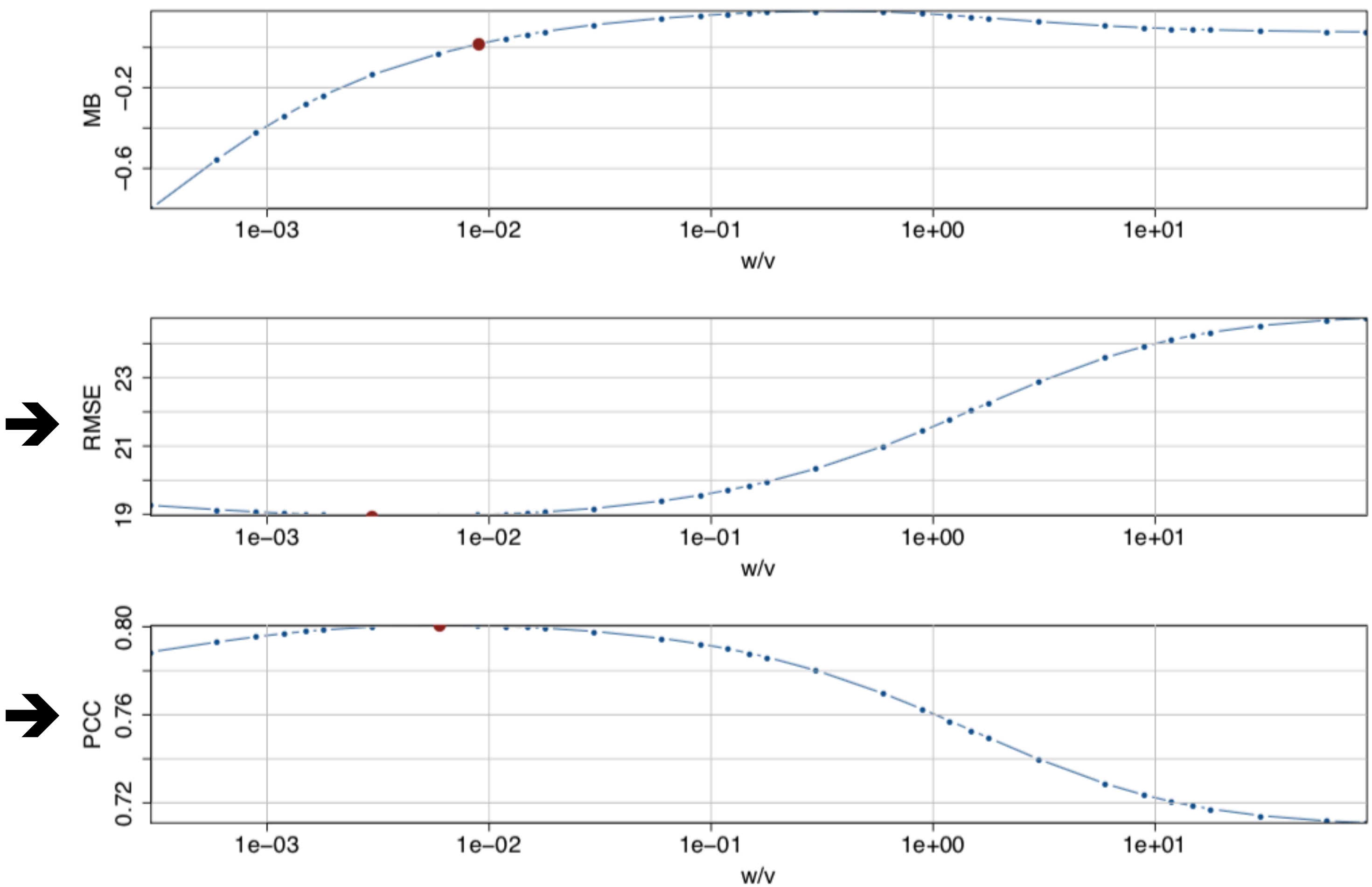
R code :

```r
if((itime+timestep) <= ntime){
    itimestep=itime%%timestep ; if(itimestep==0){itimestep=timestep}

    y_t=hdata$mod[itime]-hdata$obs[itime]
    if(is.na(y_t)==FALSE){
        k_t   <- (p_tm1_tm1[itimestep] + w_t)/(p_tm1_tm1[itimestep] + w_t + v_t)
        x_t_t <-  x_tm1_tm1[itimestep] + k_t*(y_t - x_tm1_tm1[itimestep])
        p_t_t <- (p_tm1_tm1[itimestep] + w_t)*(1 - k_t)
    }else{
        k_t   <- 0
        x_t_t <- x_tm1_tm1[itimestep]
        p_t_t <- p_tm1_tm1[itimestep] + w_t
    }

    hdata$modkf1[itime+timestep]    <- hdata$mod[itime+timestep]-x_t_t  #modkf
    hdata$corrkf1[itime+timestep]   <- x_t_t                            #corrkf
    hdata$uncertkf1[itime+timestep] <- p_t_t                            #uncertkf
    hdata$kkf1[itime+timestep]      <- k_t                              #kkf

    p_tm1_tm1[itimestep]=p_t_t
    x_tm1_tm1[itimestep]=x_t_t
}
```

Here, timestep=24 hours

NB : Possible to improve even more the filtering with lower timestep (nowcasting)

Estimation of w/v



NB : The difference of w/v ratio between the maximum of RMSE and PCC can be substantial… but the influence on the final RMSE remains quite low compared to the overall improvement obtained with KF

Illustration of the diagnostics – ES1923A(RUR)/O3

Hourly time series :

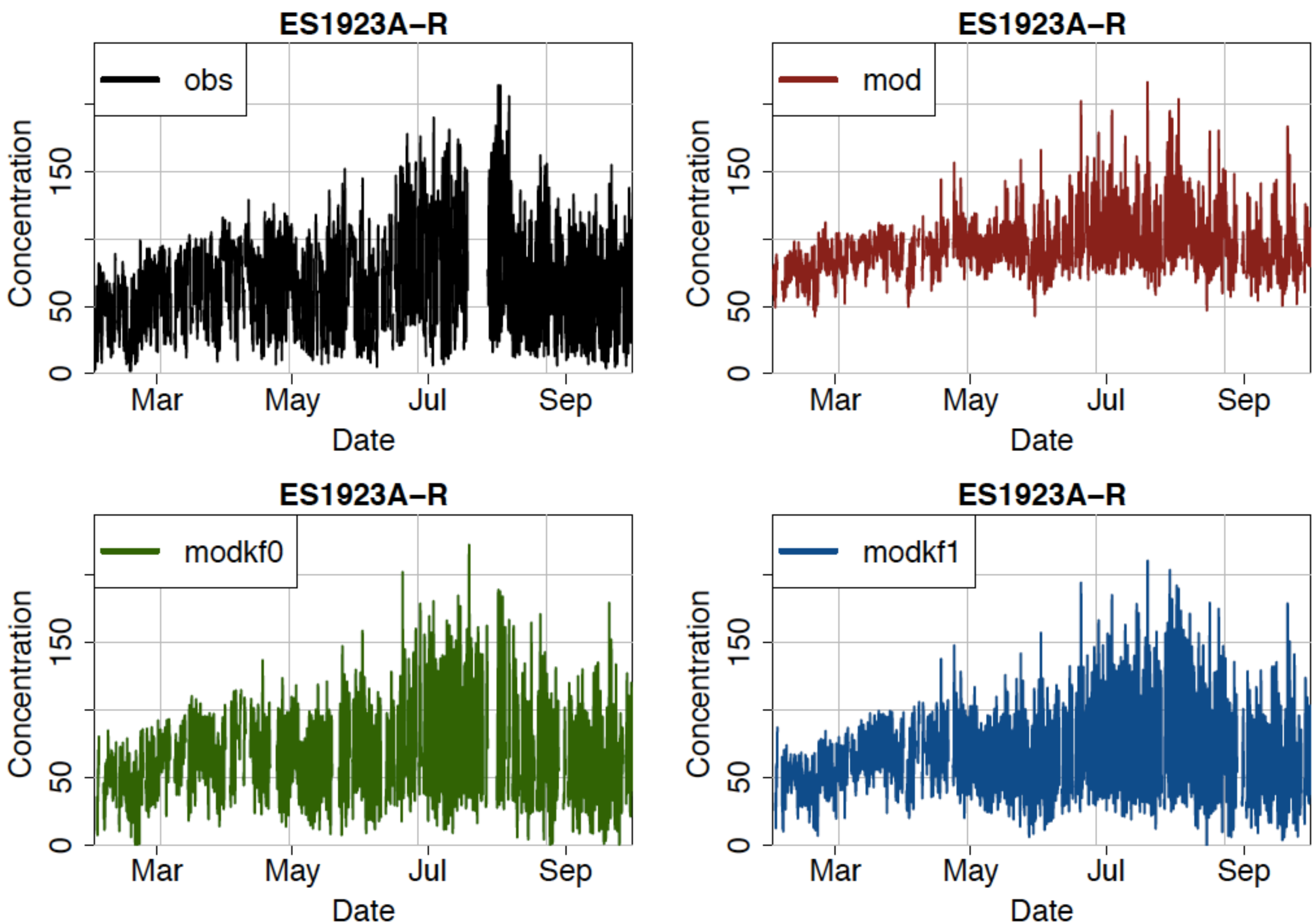
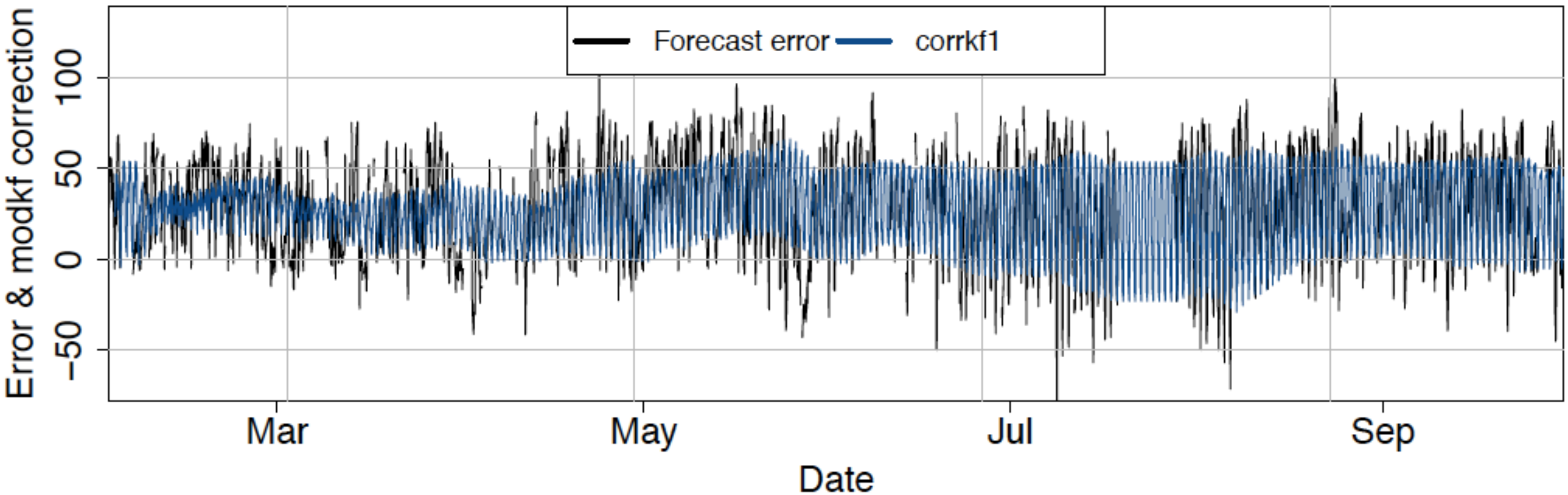


Illustration of the diagnostics – ES1923A(RUR)/O3

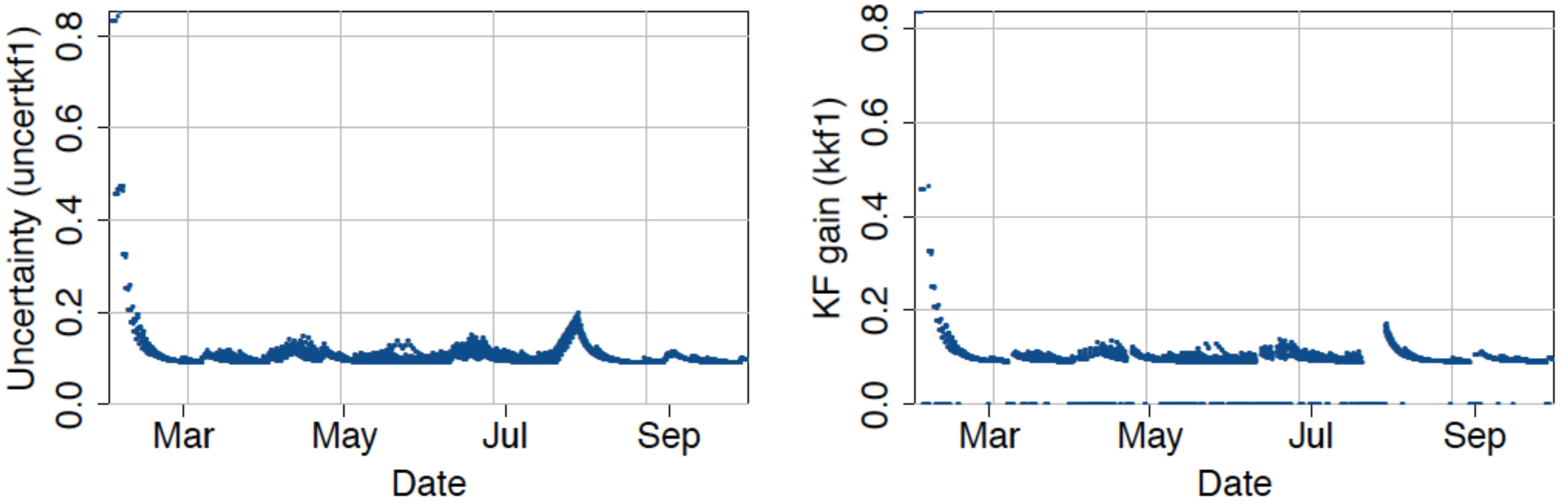
Hourly time series :



➔ KF filter unable to catch the small-scale variability of the forecast error

Illustration of the diagnostics – ES1923A(RUR)/O3

Hourly time series :

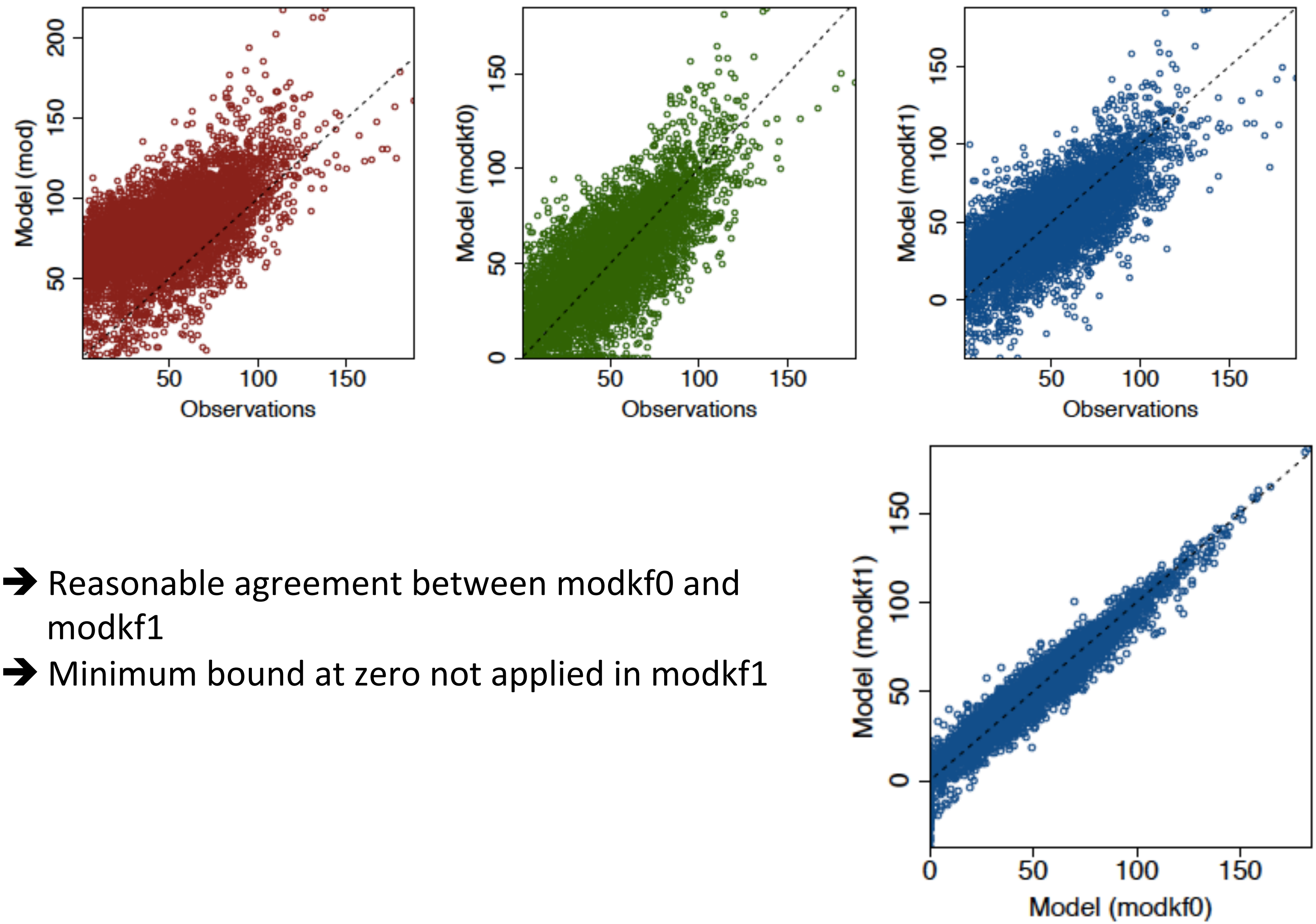


➔ Expected behavior of the KF : quick convergence (one month) of both the uncertainty and the Kalman gain to a limit value (function of the w/v ratio)

➔ When missing data : increase of the uncertainty and KF gain at zero

Illustration of the diagnostics – ES1923A(RUR)/O3

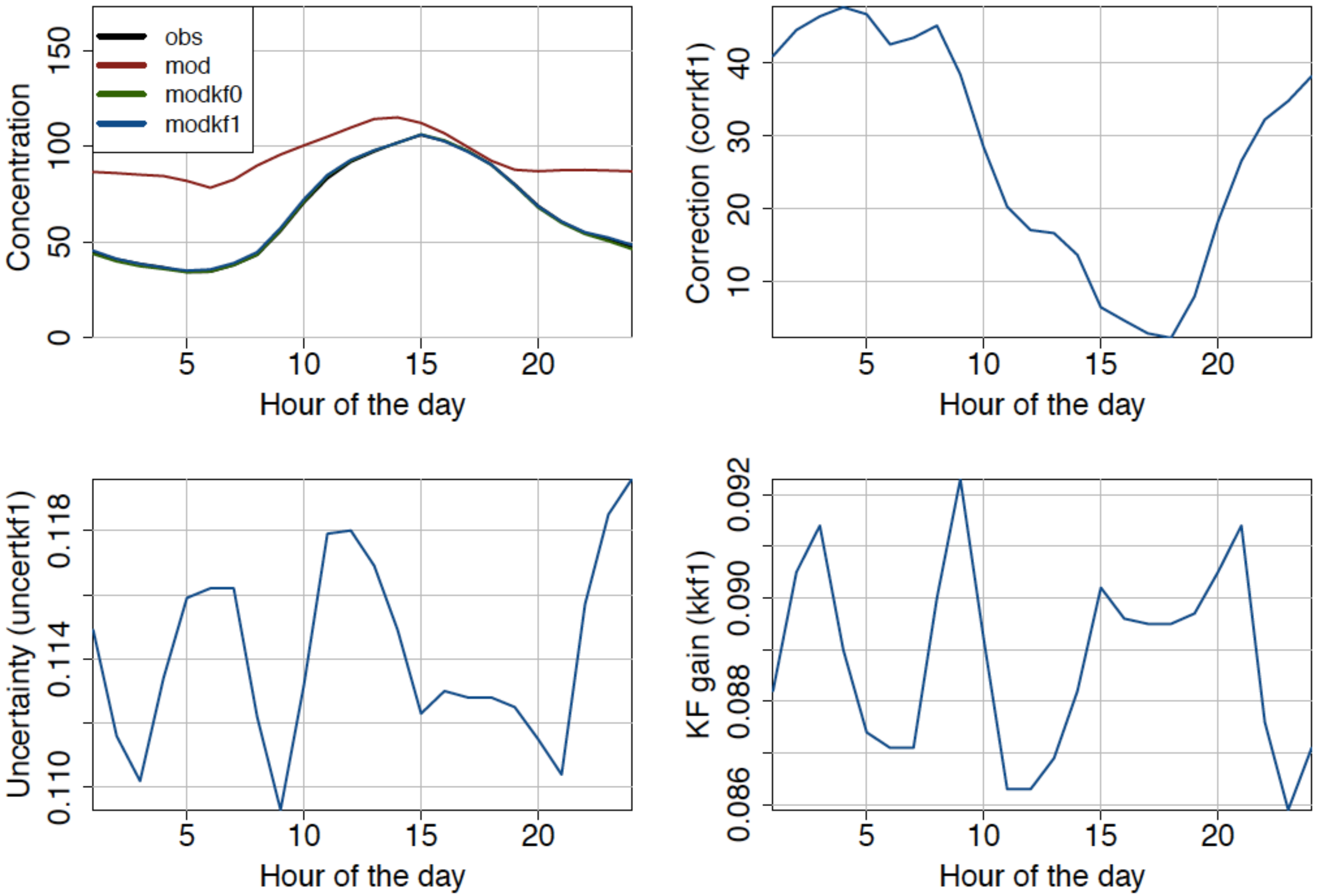
Scatter plots (hourly data) :



- ➔ Reasonable agreement between modkf0 and modkf1
- ➔ Minimum bound at zero not applied in modkf1

Illustration of the diagnostics – ES1923A(RUR)/O3

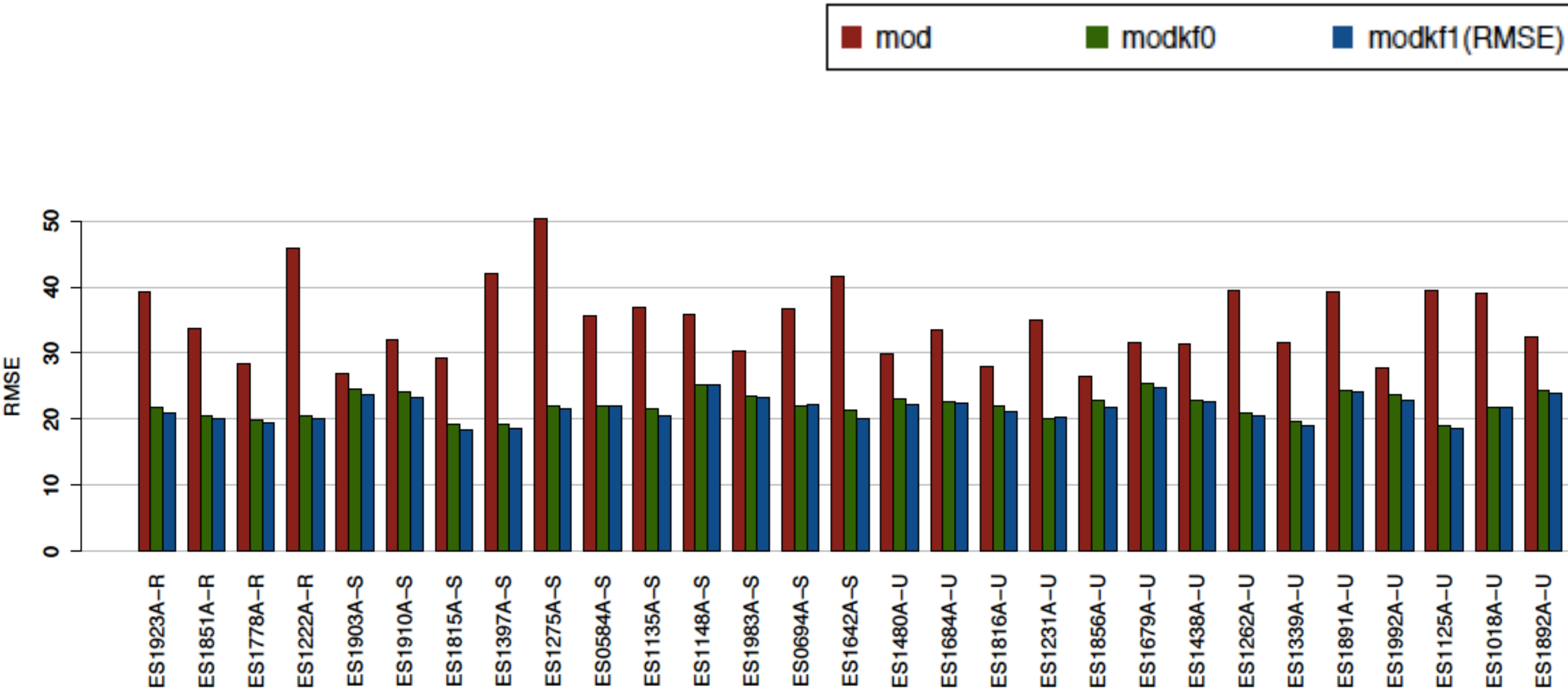
Mean diurnal profiles :



➔ Bias entirely removed by KF all along the day

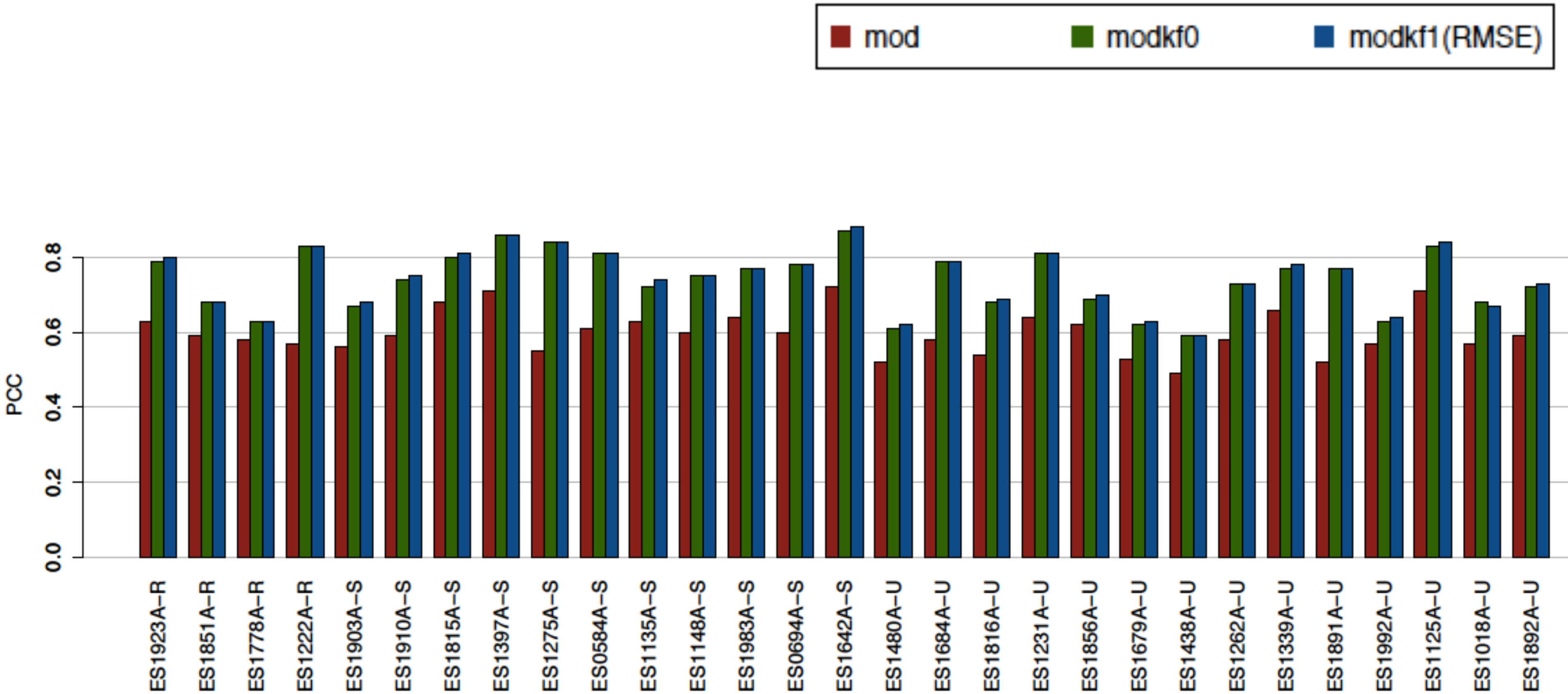
➔ Both uncertainty and KF remain roughly constant

Overview at Barcelona stations - O3



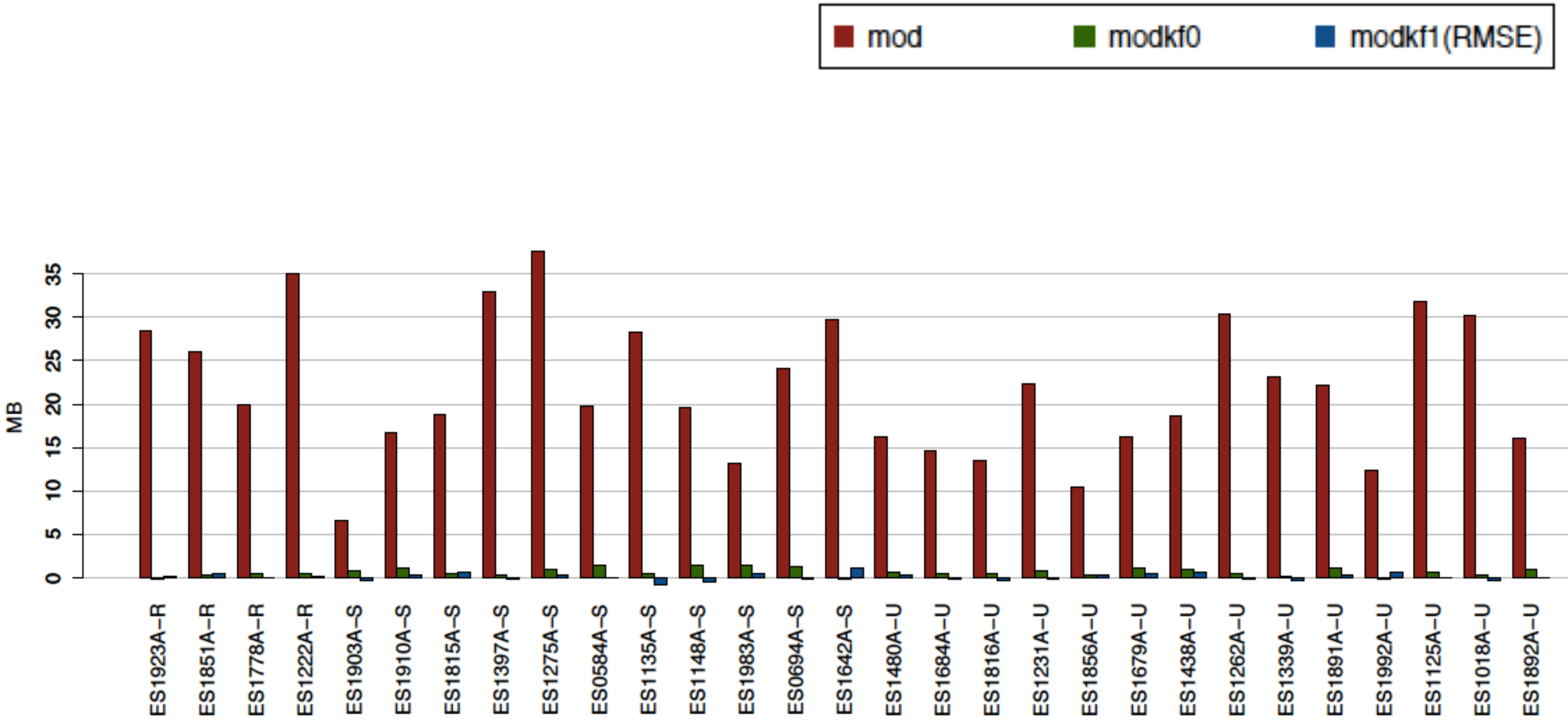
➔ Consistent results between modfk0 and modkf1 (both static and dynamic)
➔ The reduction of RMSE substantially varies from one station to the other
➔ Persistent RMSE of 15-20 ug/m3

Overview at Barcelona stations - O3



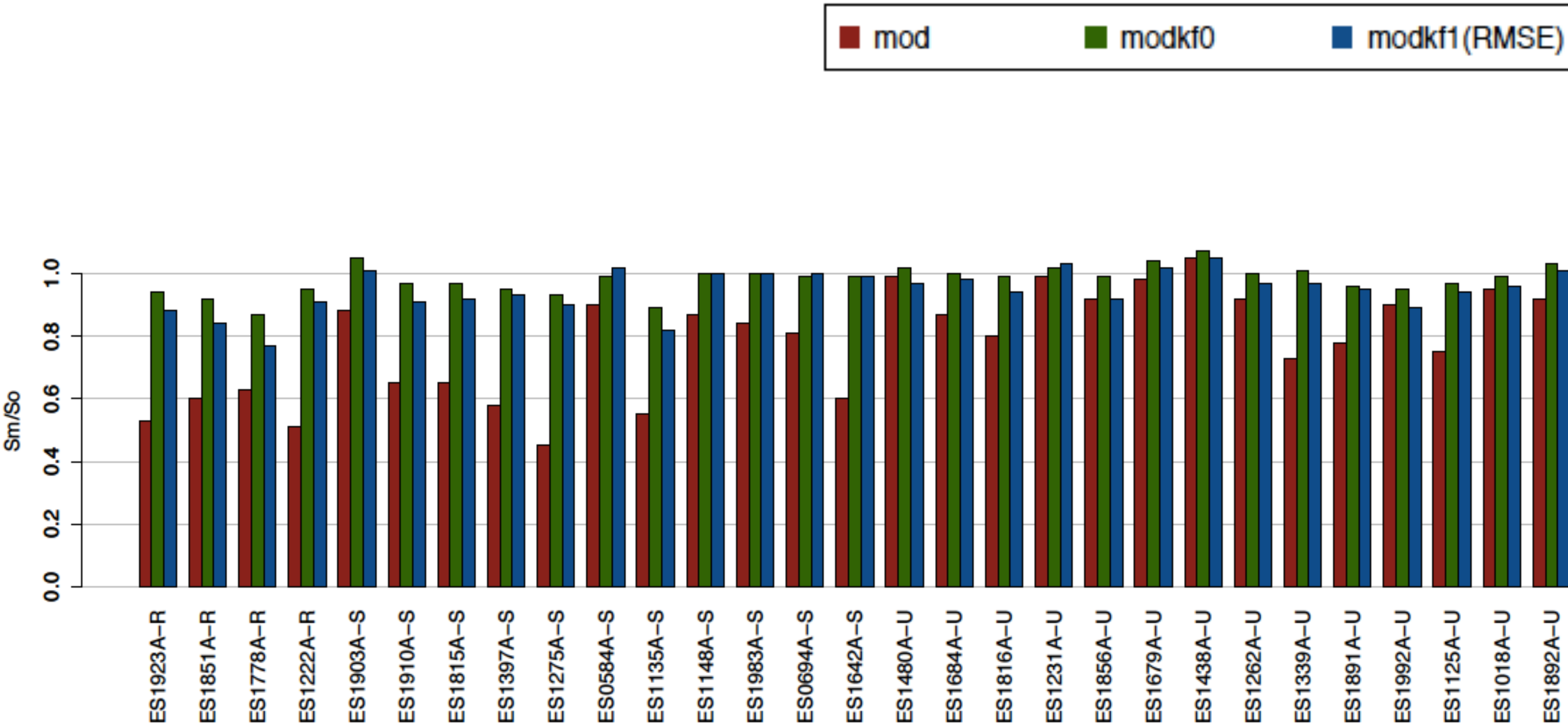
➔ Similar conclusions for the PCC (Pearson correlation coefficient)
➔ Improvement of the PCC by roughly 0.1

Overview at Barcelona stations - O3



➔ Systematic errors entirely removed at all stations

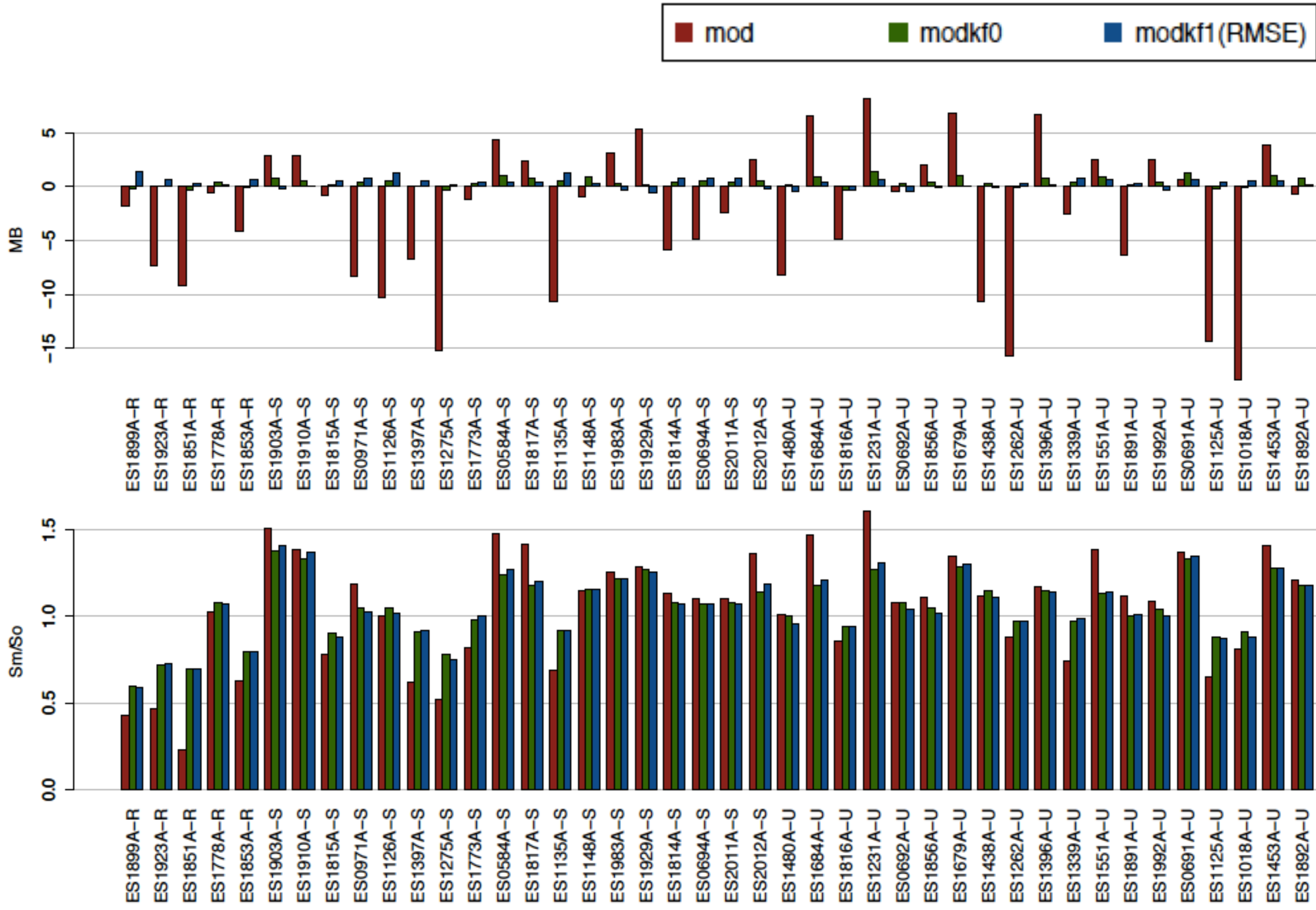
Overview at Barcelona stations - O3



➔ The (hourly) variability of O3 is underestimated by mod, which is improved with KF

Overview at Barcelona stations – NO2

Overview at Barcelona stations – NO2



Overview at Madrid stations – PM10

Overview at Madrid stations – PM10

Detection of pollution episodes : Contingency tables

mod – modkf0 – modkf1 (better/unclear/worse with the KF)

PM10 in MAD	Episode forecasted	Non-episode forecasted
Episode observed	36 – 50 - 67	109 – 67 - 78
Non-episode observed	97 – 198 - 183	6406 – 5975 - 6289

NO2 in MAD	Episode forecasted	Non-episode forecasted
Episode observed	0 – 3 - 2	31 – 28 - 29
Non-episode observed	3 – 13 - 11	225227 – 207606 - 224123

➔ An improvement on RMSE and/or PCC does not necessarily imply an improvement of the performance of the pollution episode alert system…

Detection of pollution episodes : Contingency tables

mod – modkf0 – modkf1 (better/unclear/worse with the KF)

O3 in MAD	Episode forecasted	Non-episode forecasted
Episode observed	644 – 692 - 690	550 – 425 - 504
Non-episode observed	524 – 374 - 346	5577 – 5185 - 5720

O3 in BCN	Episode forecasted	Non-episode forecasted
Episode observed	224 – 141 - 148	122 – 161 - 198
Non-episode observed	413 – 145 - 98	4816 – 4678 - 5103

➔ An improvement on RMSE and/or PCC does not necessarily imply an improvement of the performance of the pollution episode alert system…

… and results may change from one region to other

Conclusion

- The new version of the KF is consistent with the one used in the operational CALIOPE system ➔ it can be used as a reference for evaluating the performance of ML approaches

What's next?

- Kalman filter :
  - Confirm these results over the entire IP domain (544 stations) for all pollutants
  - Investigate more deeply the KF results *(e.g. spatio-temporal distribution of the bias and the KF corrections)*
  - KF with analogs? ➔ *Cf. Alicia?*

- Initiate the ML approach :
  - Build a MONARCH dataset with various features (e.g. pollutant concentrations, meteorological values, other) ➔ *Develop a tool for extracting all usefull MONARCH outputs at the location of the stations? Evaluation tool?*

  - Develop first ML approaches and compare results with KF (maybe test a few families of ML algorithms e.g. multilinear regression, tree-based models, neural networks) ➔ *Possible interactions with Leonardo Bautista Gomez and Albert Njoroge Kahira (Computer Science Department)*

Online KF (on-going work...)

Dynamic calculation of wt/vt :

In the dynamic approach, we need to estimate the values of $\mathbf{W}_t$ and $\mathbf{V}_t$. Galanis and Anadranistakis (2002) proposed to compute $\mathbf{W}_t$ and $\mathbf{V}_t$ based on the last 7 values of $\boldsymbol{\eta}_t = \mathbf{x}_t - \mathbf{x}_{t-1}$ and $\boldsymbol{\epsilon}_t = \mathbf{y}_t - \mathbf{x}_{t-1}$, respectively :

$$\begin{cases} \mathbf{W}_t = \frac{1}{6}\sum_{i=0}^{6}\left((\mathbf{x}_{t-i} - \mathbf{x}_{t-i-1}) - \frac{1}{7}\sum_{n=0}^{6}(\mathbf{x}_{t-i} - \mathbf{x}_{t-i-1})\right)^2 & (8) \\ \mathbf{V}_t = \frac{1}{6}\sum_{i=0}^{6}\left((\mathbf{y}_{t-i} - \mathbf{x}_{t-i-1}) - \frac{1}{7}\sum_{n=0}^{6}(\mathbf{y}_{t-i} - \mathbf{x}_{t-i-1})\right)^2 & (9) \end{cases}$$