Description of the package HBEF (**Hierarchical Bayes Ensemble Kalman Filter**) by

Michael Tsyrulnikov[a,*], Alexander Rakitko[a]

[a] *HydroMetCenter of Russia*

Abstract

Here, we describe the software package (in the R language) developed to conduct the numerical experiments presented in the paper **"Hierarchical Bayes Ensemble Kalman Filtering" by Michael Tsyrulnikov and Alexander Rakitko** submitted to Physica D (including the present Description as the Supplementary Material to the paper). With the package HBEF, all the results reported in the paper can be reproduced.

1. Introduction

The paper "Hierarchical Bayes Ensemble Kalman Filtering" by Michael Tsyrulnikov and Alexander Rakitko presents a new filter, Hierarchical Bayes Ensemble Kalman Filter (HBEF), designed to extend the Ensemble Kalman Filter (EnKF) for high-dimensional problems, where the background-error statistics is largely uncertain. The HBEF accommodates the conditions under which a high-dimensional EnKF actually works:

1. The ensemble size is small, so that the predictability-error covariances matrix P is unavailable.
2. The model-error covariance matrix Q is explicitly unknown.

The HBEF accounts for the uncertainty in P and Q and updates them along with the state x. In this update, ensemble members are used as generalized

[*] Corresponding author
Email address: `mik.tsyrulnikov@gmail.com` (Michael Tsyrulnikov)

observations and ordinary observations are allowed to influence the covariances.

The new filter is extensively tested in this study with a one-dimensional (scalar) model of the "truth" and synthetic observations. Thereby, the HBEF is compared with the following filters:

1. The reference Kalman Filter (KF) that "knows" the "true" model-error variance Q_k and is allowed to precisely compute the predictability-error variance P_k.
2. The variational filter (Var), where the background-error covariance matrix B_k is postulated to be constant $B_k = \bar{B}$.
3. The EnKF.
4. The predecessor of our filter, the Hierarchical Ensemble Kalman Filter (HEnKF, Myrseth and Omre, 2010).

2. Outlook

In the rest of this Supplementary Material, we outline technical details needed to run the code, briefly describe the program code, and show how it can be used to reproduce all the results presented in the paper (including numerical and graphical output of the program runs).

3. Technical details on how to interpret the code and run the program

Having installed a basic R interpreter you need to install the following standard packages:

```
library(mixAK)
library(MCMCpack)
library(pscl)
library(extRemes)
```

I.e. type `install.packages('extRemes')`.

Then, you need to include the R source file that we have developed in this study:

```
source('functions.R')
```

Note that the file `functions.R` as well as the scripts described below are to be placed in the working directory.

4. General description of the main functions and how to invoke them

Here, we outline the R functions (placed in the file 'functions.R') that comprise our package "HBEF" and describe their input and output arguments.

For brevity, we use the following informal terminology in the program code and in the present description of the R package HBEF. By a "universe", we mean a particular realization of the pseudo-random sequences $\{F_k\}$ and $\{\Sigma_k\}$. Once the "universe" is created, we may create different "worlds" in it—by generating pseudo-random sequences of the "truth" $\{x_k\}$ and observations $\{y_k\}$.

4.1. Set up parameters

Function `create_parameters_universe_world` has no input arguments.

In the function's code, one may specify/change the following basic setup parameters:

the x's mean time scale (length scale) $\bar{\tau}_x$: `tau_x`,
the F's time scale τ_F: `tau_F`,
the F's standard deviation $\mathsf{s.d.}\, F_k$: `std_F`,
the Σ's time scale τ_Σ: `tau_Sigma`,
the Σ's standard deviation $\mathsf{s.d.}\, \Sigma$: `std_Sigma`,
observation-error standard deviation $\mathsf{s.d.}\, \eta \equiv \sqrt{R}$: `std_eta`,
as well as
the number of time moments in the experiment n_{time}: `time`,
ensemble size N,
number of independent runs (worlds) L,
coefficient of Q distortion $q_{distort}$: `distort_Q`,
and four seeds for pseudo-random number generation:
seed for initiation of the $\{F_k\}$ time series `seed_for_universe1`,
seed for initiation of the $\{\Sigma_k\}$ time series `seed_for_universe2`,
seed for initiation of the pseudo-random "truth" x_k and observations y_k `seed_for_world`, and
seed for initiation of other pseudo-random sources `seed_for_filters`.

The function `create_parameters_universe_world` then calculates several internal parameters, which, together with the external ones, are encapsulated in the combined output argument `list` written, on return from the function, to the variable `parameters`.

Note that if a user performs its own experiments with our package, then, after changing one of the above parameters, he or she needs to invoke the function update_parameters in order for the changes to be in effect.

4.2. Generate the sequences $\{F_k\}$, $\{\Sigma_k\}$, and $\{Q_k\}$

Function generate_universe has parameters as the input argument. It generates pseudo-random sequences $\{F_k\}$ and $\{\Sigma_k\}$, computes $\{Q_k\}$, which we write, on return from the function generate_universe, to the variable universe.

4.3. Generate a realization of the "truth" $\{x_k\}$ and observations $\{y_k\}$

Function generate_world has parameters and universe as the input arguments. It generates pseudo-random sequences $\{x_k\}$ and $\{y_k\}$, which we write, on return from the function generate_world, to the variable world.

4.4. Run the KF

Function filter_kf has world, universe, parameters, parameters_kf as the input arguments. The KF-specific input variable parameters_kf contains B_f_0 used as B_0 to start the filter.

The invoking script writes the output of the function filter_kf to the variable output_kf, which contains the time series (sequences) of:

deterministic background forecasts x_k^f,

deterministic analyses x_k^a,

(prior) background-error variances B_k^f,

posterior background-error variances equal, for the KF, to the prior ones: $B_k^a = B_k^f$,

and

analysis-error variances A_k.

4.5. Run the Var

Function filter_var has world, universe, parameters, parameters_var as the input arguments. The Var-specific input variable parameters_var contains mean_B ($\bar{B}$) used as the constant background-error variance $\bar{B}$.

The invoking script writes the output of the function filter_var to the variable output_var, which contains the time series (sequences) of:

deterministic background forecasts x_k^f,

deterministic analyses x_k^a,

prior background-error variances, which, for the Var, are constant: $B_k^f = \bar{B}$,

and

posterior background-error variances equal, which, for the Var, are, again, constant: $B_k^f = \bar{B}$.

4.6. Run the EnKF

Function `filter_enkf` has `world, universe, parameters, parameters_enkf` as the input arguments. The EnKF-specific input variable `parameters_enkf` contains the variance inflation parameter `inflation` used to multiply the background-ensemble perturbations.

The invoking script writes the output of the function `filter_enkf` to the variable `output_enkf`, which contains the time series (sequences) of:

deterministic background forecasts x_k^f,

deterministic analyses x_k^a,

prior background-error variances B_k^f,

and

posterior background-error variances equal, for the EnKF, to the prior ones: $B_k^a = B_k^f$.

4.7. Run the HEnKF

Function `filter_henkf` has `world, universe, parameters, parameters_henkf` as the input arguments. The HEnKF-specific input variable `parameters_henkf` contains `mean_B` ($\bar{B}$) used to start the filter and the Inverse Gamma dispersion parameter `theta` (θ) used to define the prior for B_k.

The invoking script writes the output of the function `filter_henkf` to the variable `output_henkf`, which contains the time series (sequences) of:

deterministic background forecasts x_k^f,

deterministic analyses x_k^a,

prior background-error variances B_k^f,

and

posterior background-error variances B_k^a.

4.8. Run the HBEF

Function `filter_hbef` has `world, universe, parameters, parameters_hbef` as the input arguments. The HBEF-specific input variable `parameters_hbef` contains:

the dispersion parameter for the Inverse Gamma distribution $Q|Q^f$: χ,

the dispersion parameter for the Inverse Gamma distribution $\Pi|\Pi^f$: ϕ,

the dispersion parameter for the Inverse Gamma distribution $P|\Pi$: θ,

size of the Monte-Carlo sample used to estimate the posterior mean $\bar{m}^a_{MC}$: `size_for_MC`,

starting value for the analysis-error variance A: `mean_A`,

starting value for the model-error variance Q: `mean_Q`, and

the logical switch controlling whether the approximated posterior is to be used (=TRUE if yes): `approximation`, and

the logical switch controlling whether the factor $L_o(B)$ is used to compute the posterior, i.e. whether observations y are allowed to influence the variances P and Q (=TRUE if yes): `use_L_o`.

The invoking script writes the output of the function `filter_hbef` to the variable `output_hbef`, which contains the time series (sequences) of:

deterministic background forecasts x^f_k,

deterministic analyses x^a_k,

prior model-error variances Q^f_k,

posterior model-error variances Q^a_k,

prior predictability-error variances Π^f_k,

posterior predictability-error variances P^a_k,

prior background-error variances $B^f_k = \Pi^f_k + Q^f_k$,

and

posterior background-error variances $B^a_k = P^a_k + Q^a_k$.

5. Numerical experiments: "technology"

In the following sections, we outline the R code that enables the reproduction of all numerical experiments presented in the paper. Then, for each experiment, we give the experimental results that are to be reproduced if the user runs the program in the default setting. If the output we give coincides with that obtained by the user, then everything is OK and the user can change the setup parameters and run other experiments.

The user can also launch one single script `RunAll.R`, which invokes all other scripts and computes all the results presented in the paper. Note that the script `RunAll.R` takes several hours to perform all computations.

5.1. Computing and storing data needed to estimate the "true" background-error variances B_k (for all filters), the variances of the "truth" V_k, and the filters' outputs

Run the script `Calculate_data_for_B_evaluation.R`. Before running the script, you may change the number of time steps `parameters$time` and the number L of independent realizations (assimilation runs) `parameters$L`. As a result of an execution of the script, you may see the appearance, in the working directory, of 10 new data files like `X_true`, `B_a_hbef`, etc.

We recommend to run this script first because a number of other scripts use its output (as indicated in each particular case below and in the header comments of the respective script).

5.2. Calculate RMSEs of the analyses for all filters

Just run the script `RMSE.R`.

The output should be as follows:

```
 Filter   RMSE
KF        2.467382
Var       2.679803
EnKF      2.644797
HEnKF     2.633884
HBEF      2.548824
```

5.3. Plot a segment of the time series of the "truth" and its filters' estimates

Plot a segment of the time series of: the "truth" along with the analyses of KF, EnKF, and HBEF.

Run the script `Filters_plot.R`. Before running the script, you may change the number of time moments in the time series, `parameters$time` and select the segment to be plotted, `t1,t2` (within the range from 1 to `parameters$time`).

The output should be as in Figs.1 and 2 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.7.

5.4. Compute RMSEs for the state x as functions of N

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the ensemble size N.

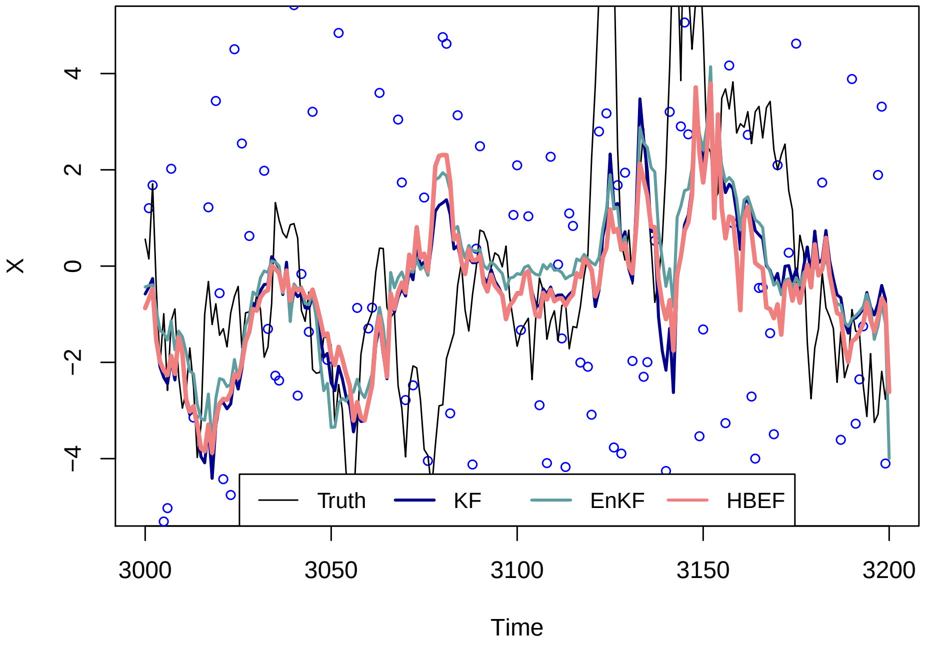

Figure 1: Time series for: “truth”, observations (circles), and deterministic analyses of the filters: KF, EnKF, and HBEF.

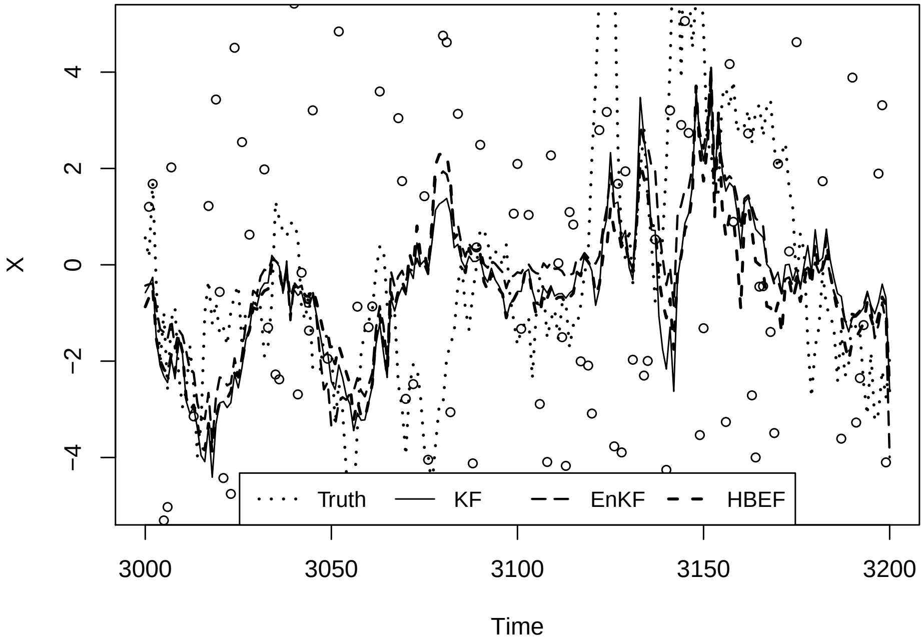

Figure 2: Time series for: “truth”, observations (circles), and deterministic analyses of the filters: KF, EnKF, and HBEF.

Run the script `RMSE_N.R`. Before running the script, you may change the range of N for which the computations are to be performed, `range`.

The output should be as in Figs.3 and 4 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.8(top, left).

5.5. Compute and plot RMSEs for the state x as functions of $\sqrt{R}$

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the observation-error standard deviation $\sqrt{R}$.

Run the script `RMSE_R.R`. Before running the script, you may change the values of $\sqrt{R}$ for which the computations are to be performed, `range`.

The output should be as in Figs.5 and 6 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.8(top, right).

5.6. Compute and plot RMSEs for the state x as functions of π

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the degree of local instability of the system measured by the probability π of the event $|F_k| > 1$: $\pi = P(|F_k| > 1)$.

Run the script `RMSE_pi.R`. Before running the script, you may change the values of π for which the computations are to be performed, `range`.

The output should be as in Figs.7 and 8 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.8(bottom, left).

5.7. Compute and plot RMSEs for the state x as functions of `s.d.` Σ

Compute and plot the RMSEs for the state x for the KF, Var, EnKF, HEnKF, and HBEF – as functions of the degree of variability of the system-noise (model error) measured by the st.dev of Σ.

Run the script `RMSE_sdSigma.R`. Before running the script, you may change the values of `s.d.` Σ for which the computations are to be performed, `range`.

The output should be as in Figs.9 and 10 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.8(bottom, right).

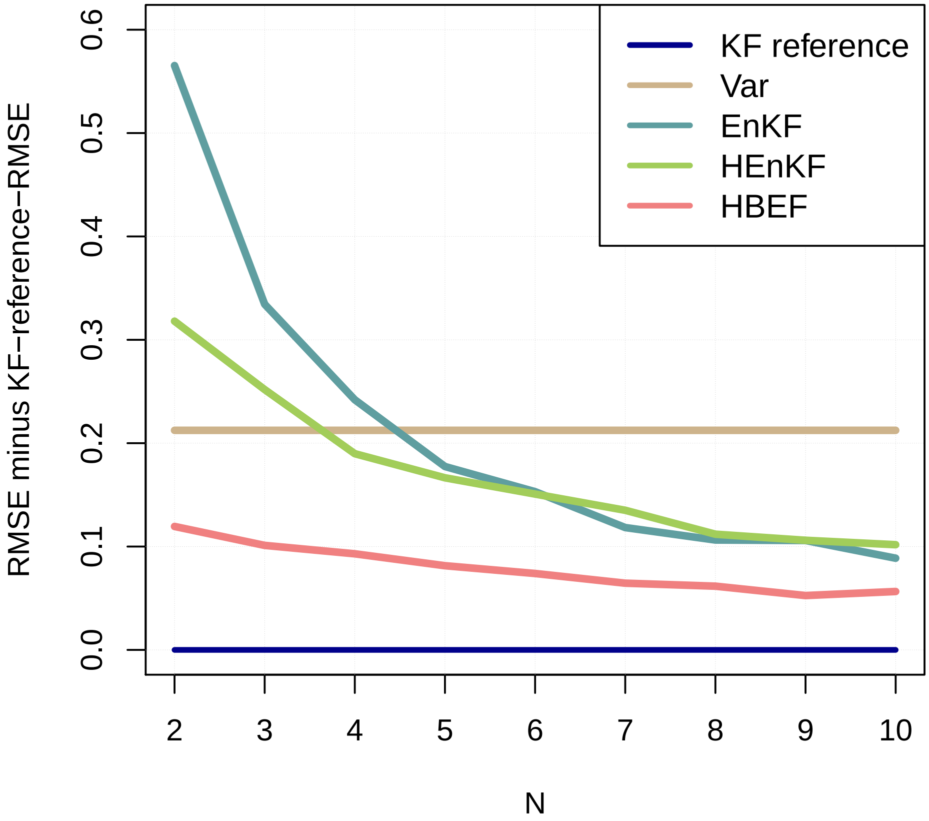

Figure 3: RMSEs as functions of N

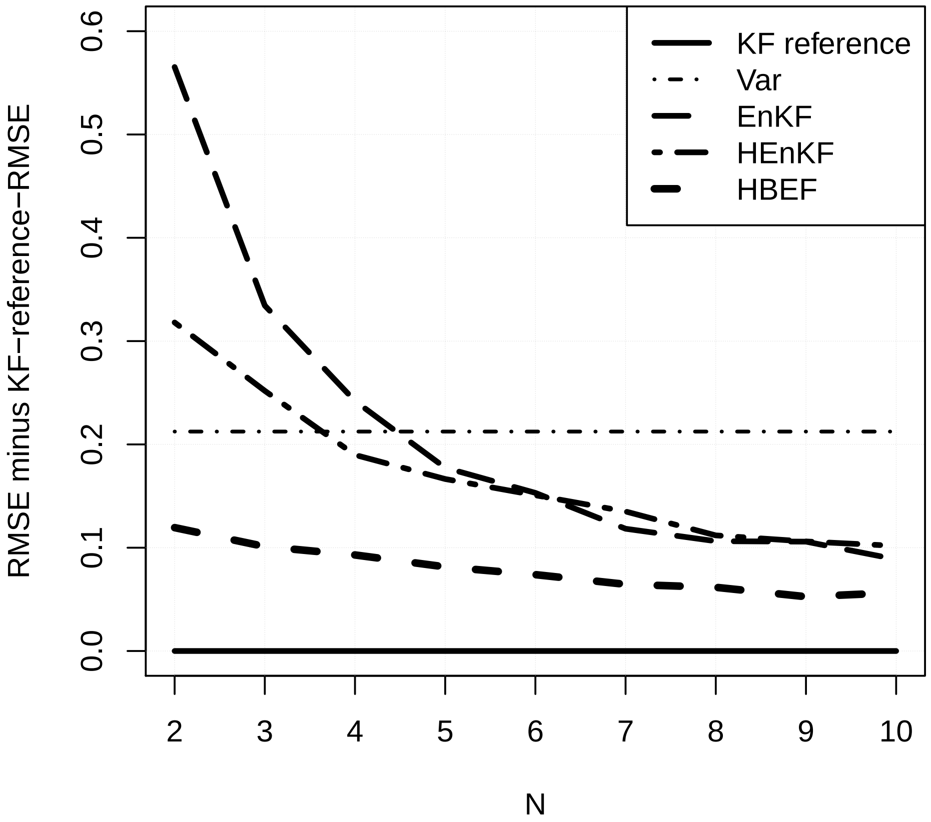

Figure 4: RMSEs as functions of N

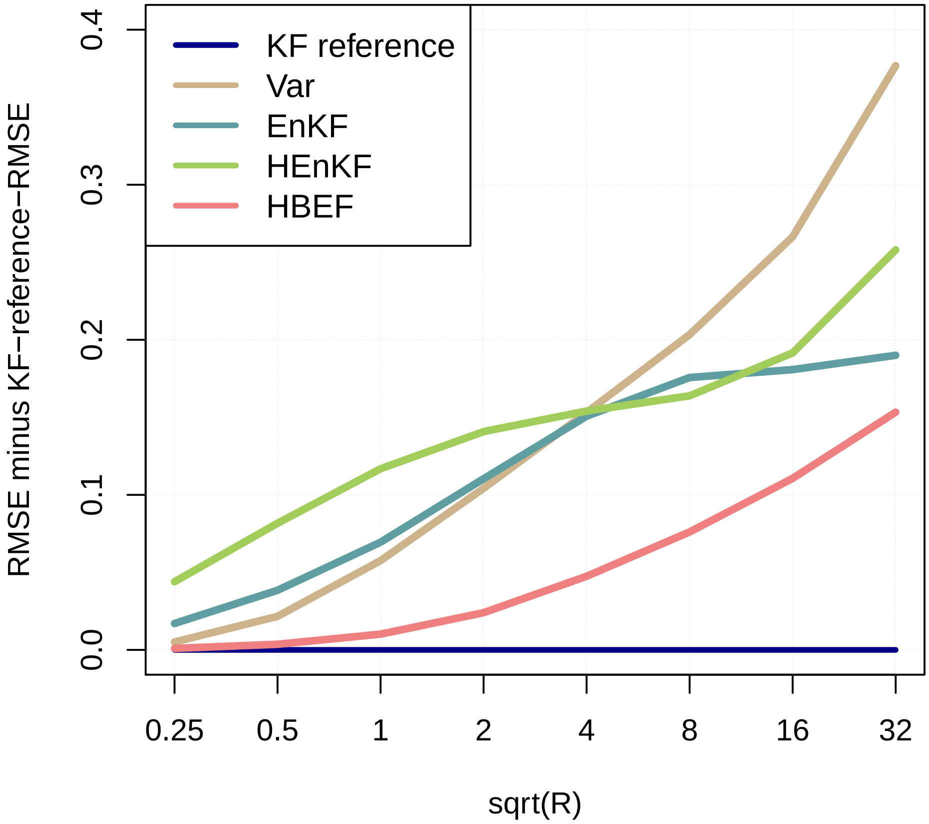

Figure 5: RMSEs as functions of $\sqrt{R}$

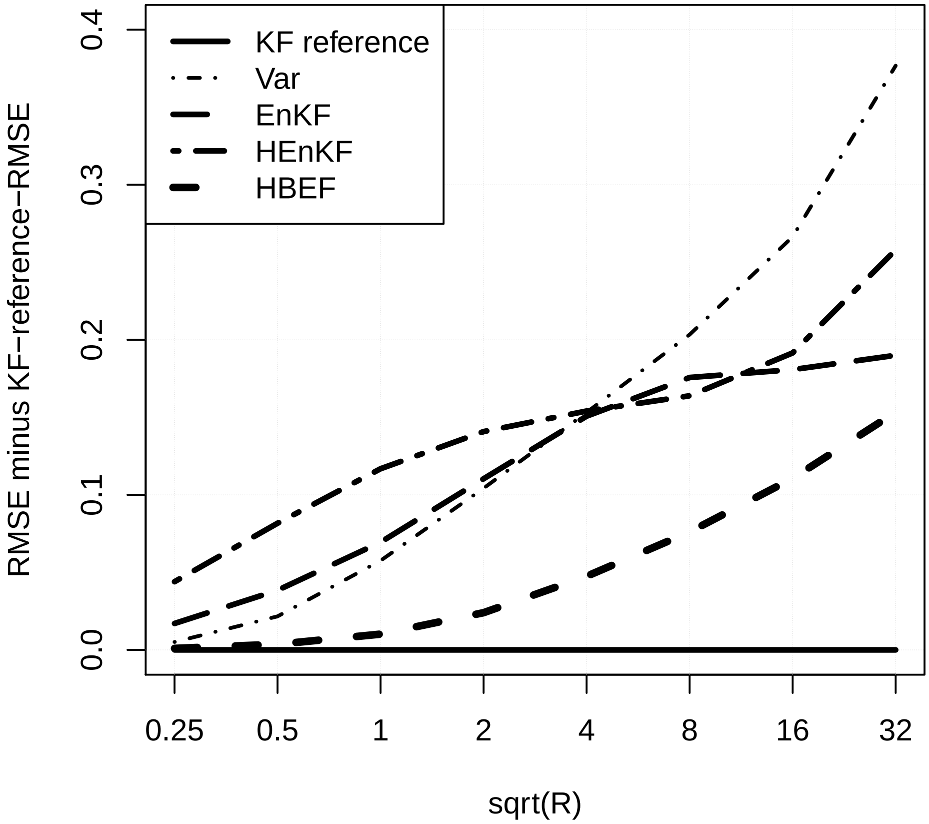

Figure 6: RMSEs as functions of $\sqrt{R}$

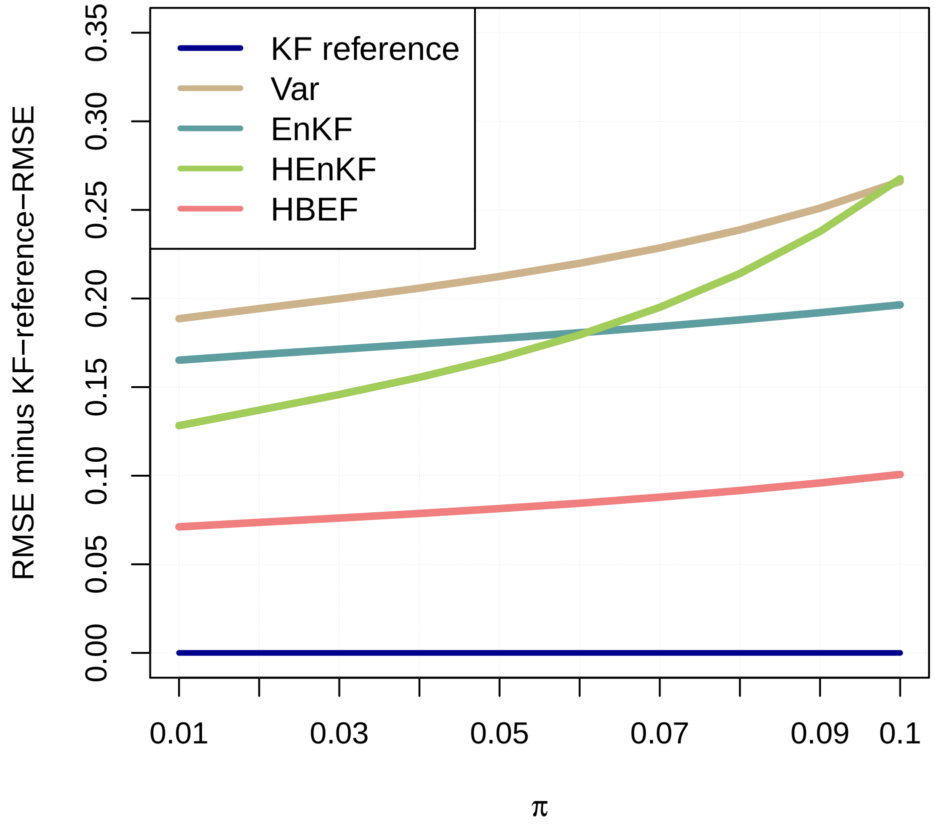

Figure 7: RMSEs as functions of π

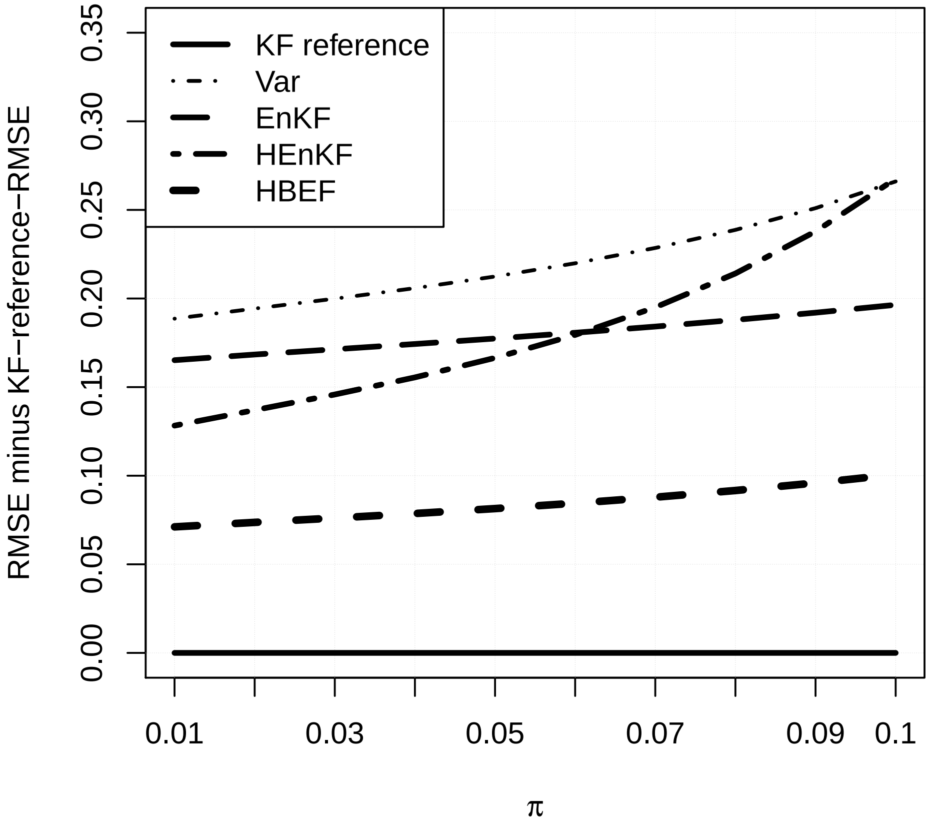

Figure 8: RMSEs as functions of π

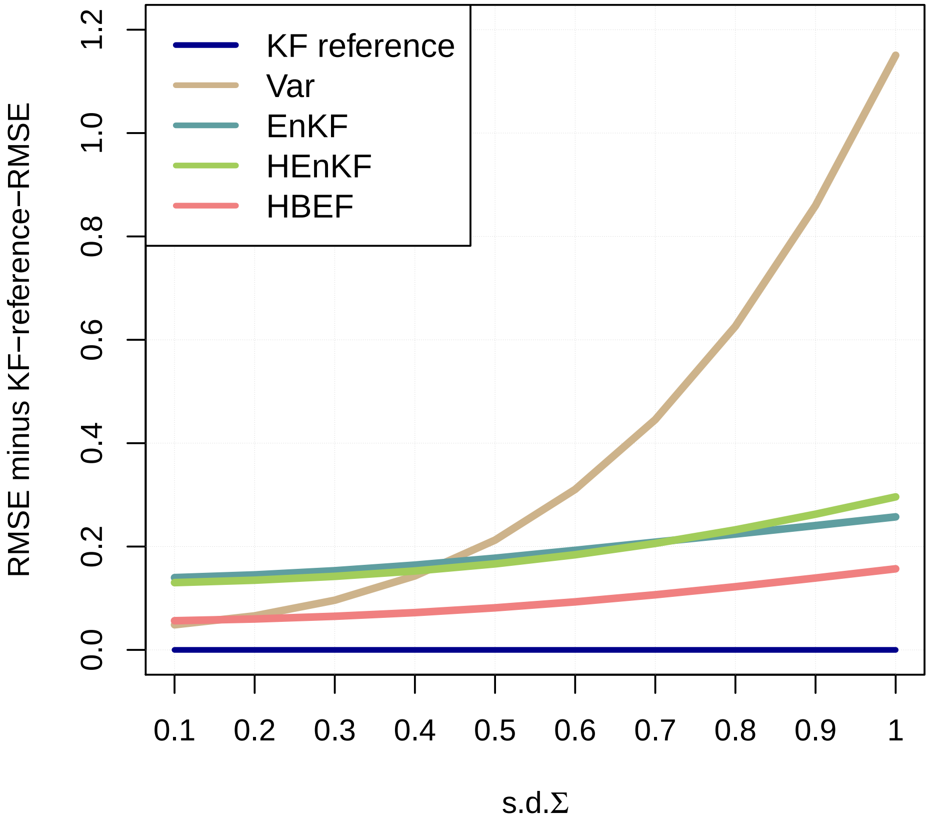

Figure 9: RMSEs as functions of $\mathsf{s.d.}\,\Sigma$

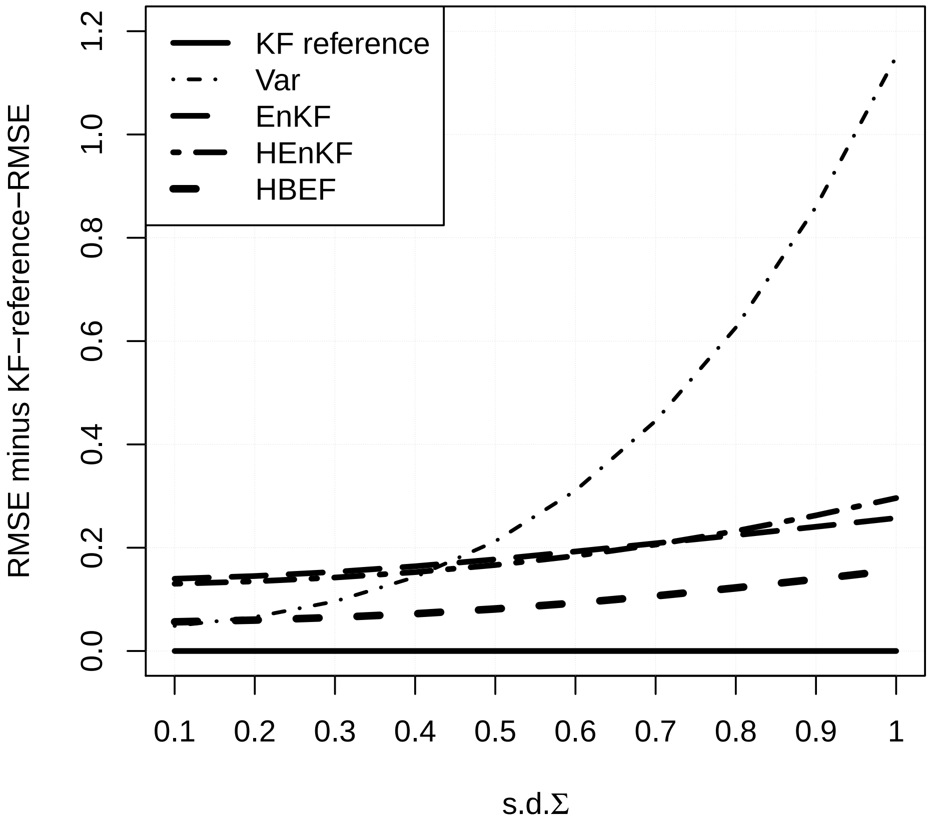

Figure 10: RMSEs as functions of s.d. Σ

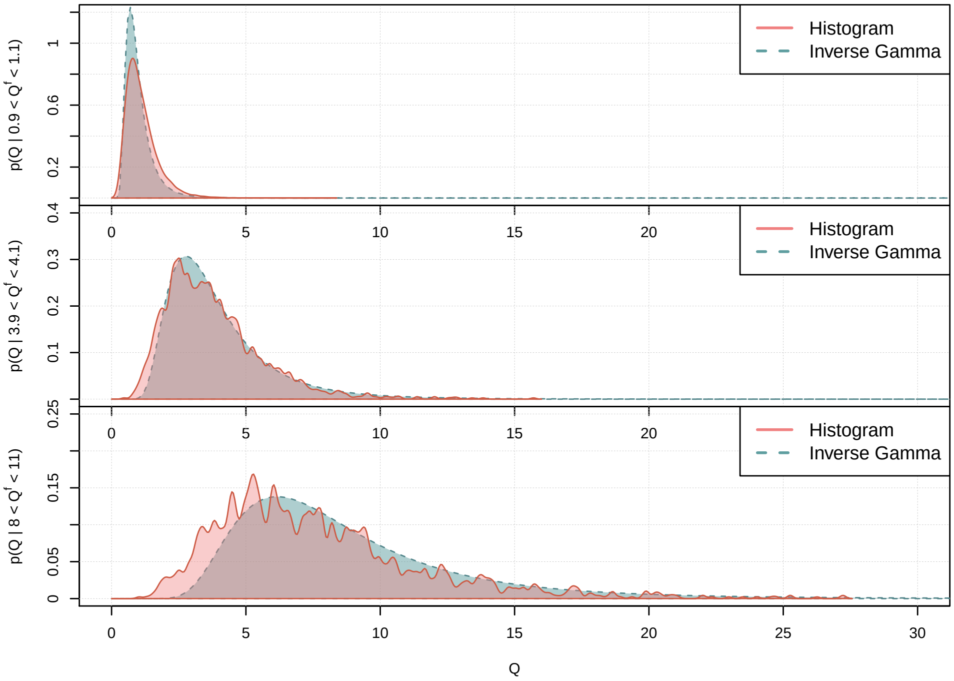

Figure 11: Histogram and inverse-Gamma approximation for $Q|Q^f$

5.8. Compute and plot the histogram and the approximating Inverse Gamma pdf for the distribution $Q|Q^f$

Run the script `dens_Q.R`. Before running the script, you may change the intervals, where the conditioning Q^f lies: (`bot_bound` , `up_bound`) and also change the number of the intervals, `num_of_plots`.

The output should be as in Figs.11 and 12 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.2.

5.9. Compute and plot the histogram and the approximating Inverse Gamma pdf for the distribution $\Pi|\Pi^f$

Run the script `dens_Pi.R`. Before running the script, you may change the intervals, where the conditioning Π^f lies: (`bot_bound` , `up_bound`) and also change the number of the intervals, `num_of_plots`.

The output should be as in Figs.13 and 14 in this supplementary text (in color and in black-and-white).

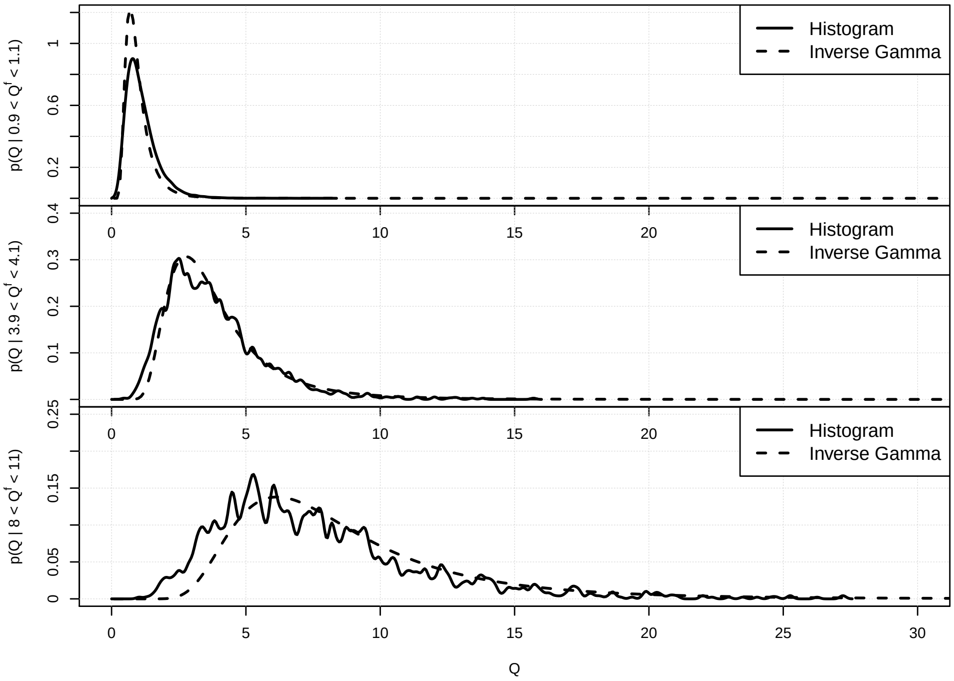

Figure 12: Histogram and inverse-Gamma approximation for $Q|Q^f$

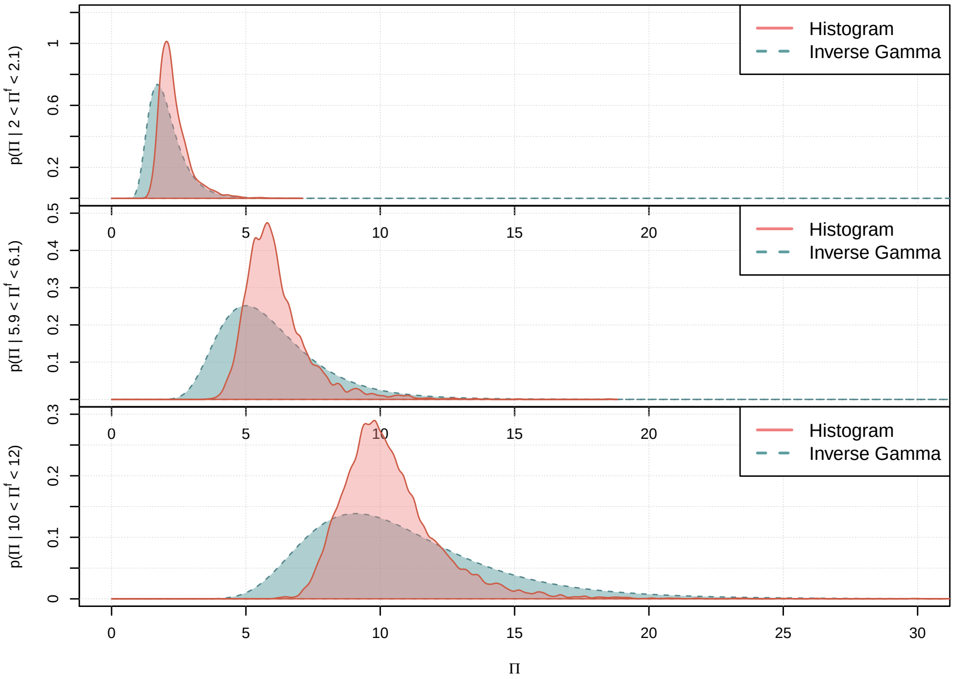

Figure 13: Histogram and inverse-Gamma approximation for $\Pi|\Pi^f$

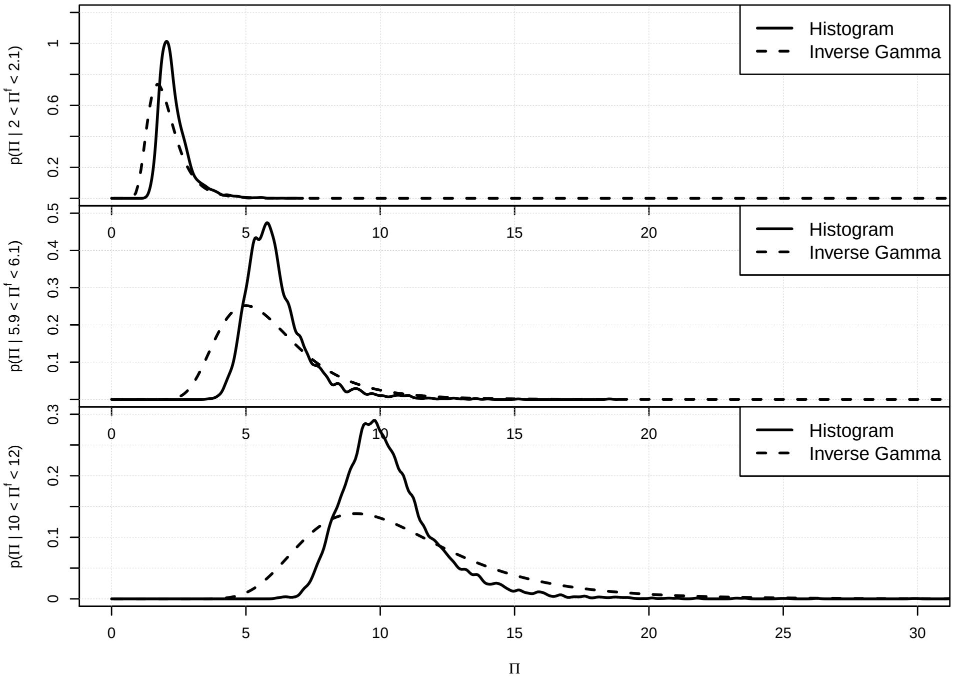

Figure 14: Histogram and inverse-Gamma approximation for $\Pi|\Pi^f$

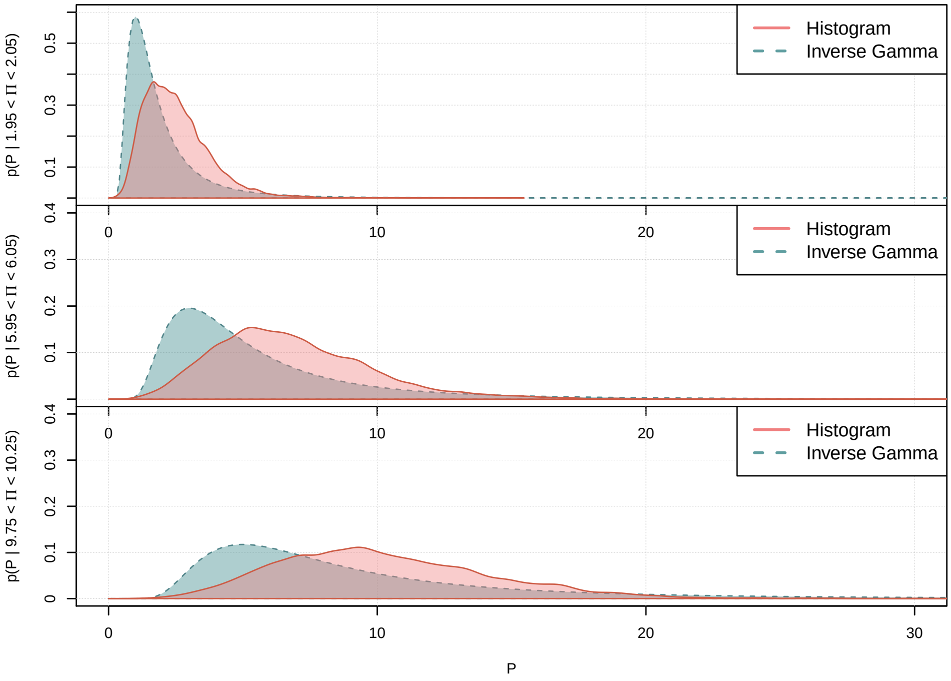

Figure 15: Histogram and inverse-Gamma approximation for $P|\Pi$

In the paper, this is Fig.3(left).

5.10. Compute and plot the histogram and the approximating Inverse Gamma pdf for the distribution $P|\Pi$

Run the script `dens_P.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `dens_P.R`, you may change the intervals, where the conditioning Π lies: (`bot_bound` , `up_bound`) and also change the number of the intervals, `num_of_plots`.

The output should be as in Figs.15 and 16 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.3(right).

5.11. Estimation of the variances and their error statistics

The script `Evaluate_B.R` estimates the "true" background-error variances B_k for each filter separately, the variances of the "truth" V_k and com-

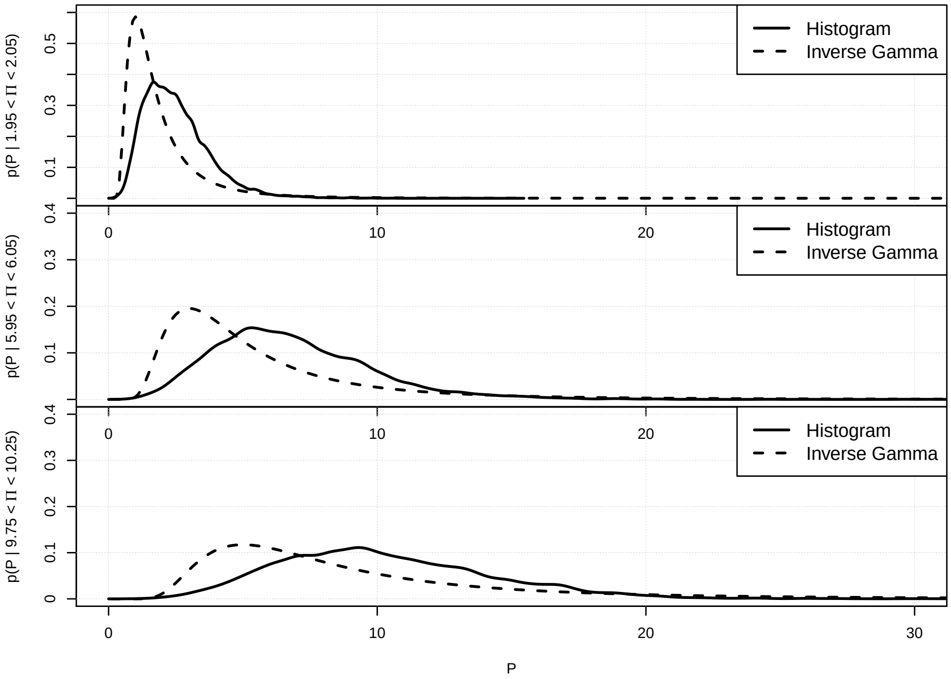

Figure 16: Histogram and inverse-Gamma approximation for $P|\Pi$

putes the error statistics for B_k: bias and RMS of the predicted by the filters B w.r.t. the "true" one.

Execution: run the script `Evaluate_B.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `Evaluate_B.R`, you may change the number of time steps `parameters$time` and the number L of independent realizations (assimilation runs) `parameters$L`. As a result of an execution of the script, you should obtain the following statistics:

```
       Mean(B_hat–B)  RMSE(B_hat–B)   Mean(B)
KF        0.00363446      0.8953872  7.020340
Var      -0.90820962      7.1083990  7.932184
EnKF     -0.75711330      6.8316686  7.865216
HEnKF    -3.46824215      5.8791197  7.887316
HBEF     -0.58355820      3.9484903  7.395490
```

Here `B_hat` stands for B^* encountered in the paper. These statistics appears in Table 1 in the body of the paper.

5.12. Plot a segment of the time series of $F, \sigma, V, B, B - B^{KF}$

Plot a segment of the time series of: the model's operator F_k, the system-noise standard deviation σ_k, the variance of the "truth" V_k, the estimated "true" background-error variance for the HBEF B_k, and the differences of the estimated "true" background-error variances for the HBEF and the EnKF with the background-error variance for the exact (reference) KF.

Run the script `Timeseries.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `Timeseries.R`, you may change the number of time moments in the time series, `parameters$time` and select the segment to be plotted, `t1,t2` (within the range from 1 to `parameters$time`).

The output should be as in Figs.17 and 18 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.1.

5.13. Quantile-quantile (qq) plot for the unconditional prior distribution of the state

Compute the $q - q$ (quantile-quantile) plot, which reflects the degree of Gaussianity, and the Gaussian (normal) approximation for the for the unconditional background-error distribution $p(x - m^f)$.

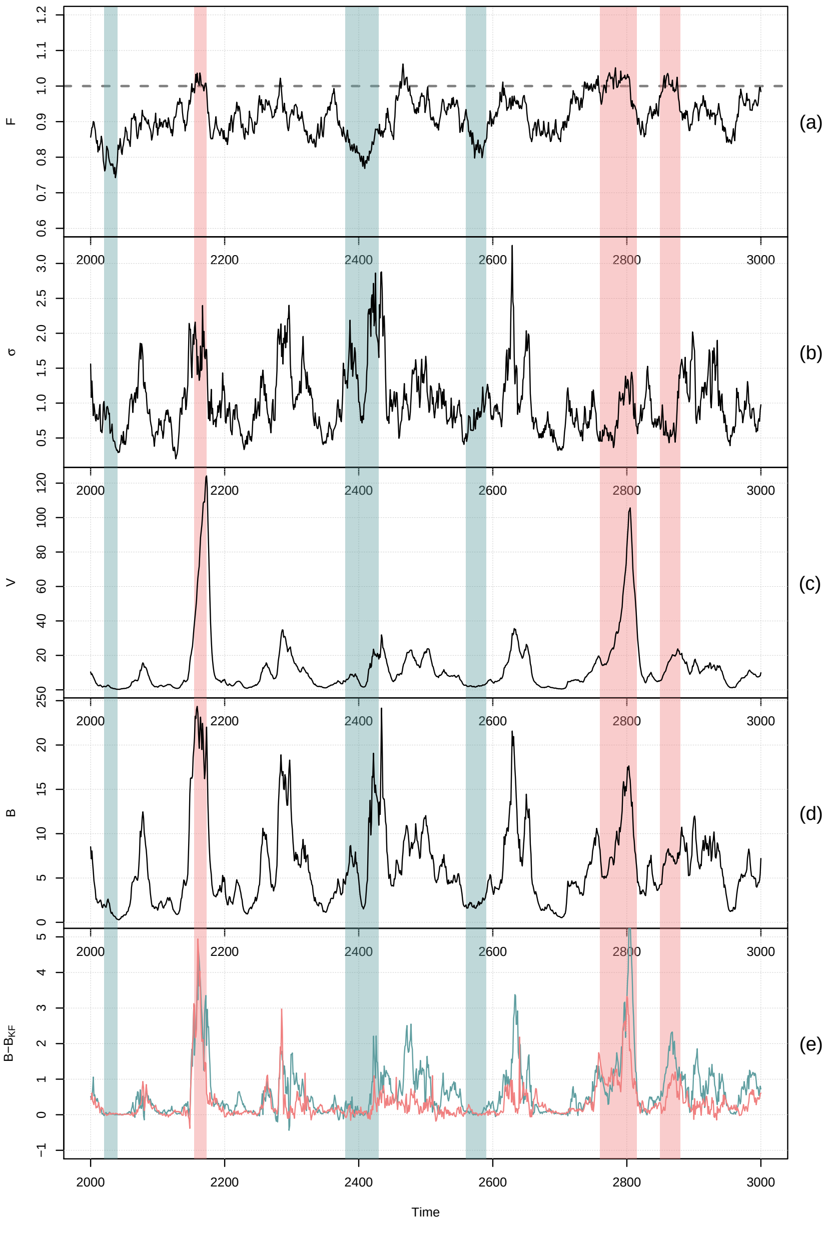

Figure 17: Typical time series of: (a) Forecast operator F_k, (b) Model-error standard deviation σ_k, (c) Variance of "truth" V_k, (d) Background-error variance B_k for HBEF, and (e) Background-error variances for HBEF (pink) and EnKF (blue) with the reference-KF background-error variance subtracted.

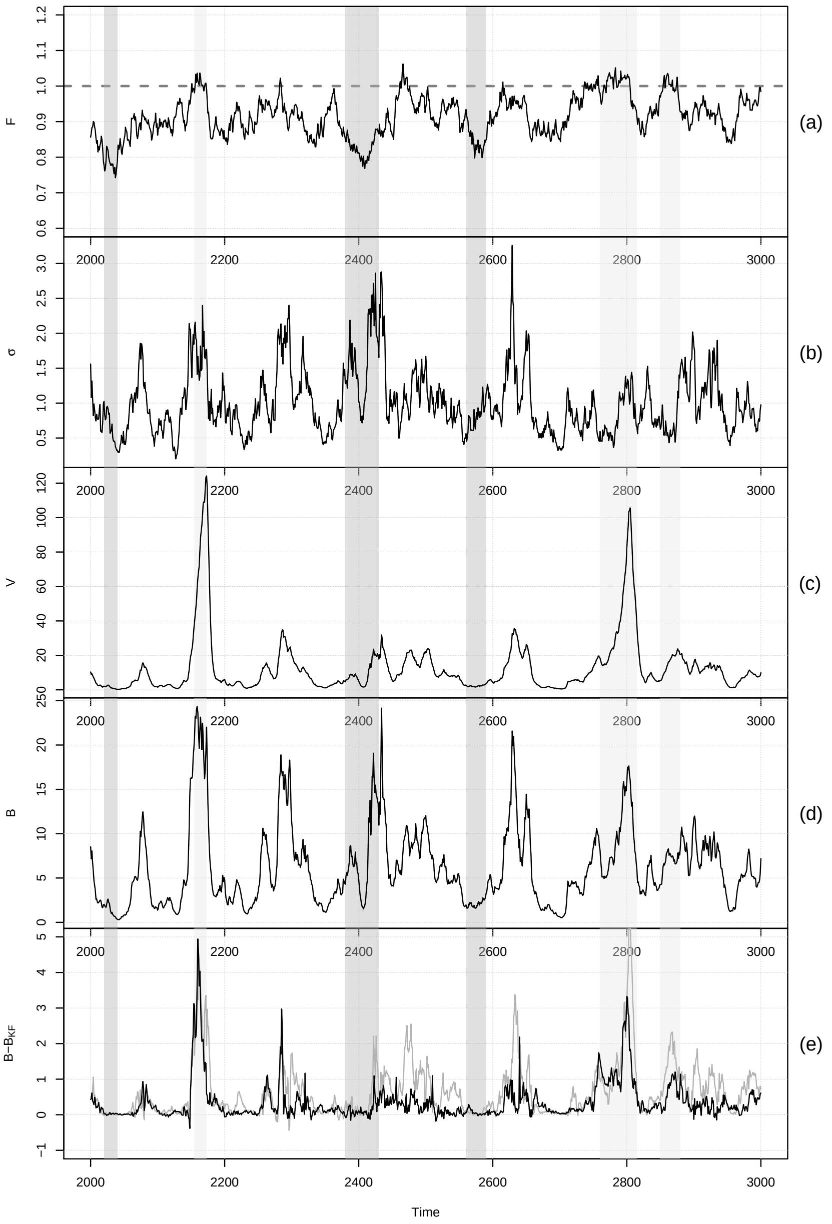

Figure 18: Typical time series of: (a) Forecast operator F_k, (b) Model-error standard deviation σ_k, (c) Variance of “truth” V_k, (d) Background-error variance B_k for HBEF, and (e) Background-error variances for HBEF (black) and EnKF (light gray) with the reference-KF background-error variance subtracted.

Run the script `qqplot_1.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `qqplot_1.R`, you may change the sample size parameters: `parameters$time` and `parameters$L`.

The output should be as in Figs.19 and 20 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.4(right).

5.14. Quantile-quantile (qq) plots for the conditional prior distribution of the state

Compute the $q - q$ (quantile-quantile) plots, and the Gaussian (normal) approximations for the conditional background-error distribution $p(x - m^f|B)$.

Run the script `qqplot_2.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `qqplot_2.R`, you may change the sample size parameters: `parameters$time` and `parameters$L`.

The output should be as in Figs.21 and 22 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.4(left).

5.15. Time series for $Q^f - Q$ and $S^{me} - Q$

Plot a segment of the time series of $Q^f - Q$ and $S^{me} - Q$.

Run the script `QfSme_timser.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `QfSme_timser.R`, you may change the sample size parameters: `parameters$time`, `parameters$L`, and the segment of the time series `t1, t2` such that 1 < `t1` < `t2`< `parameters$time`.

The output should be as in Figs.23 and 24 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.5(left).

5.16. Time series for $\Pi^f - P$ and $S^{pe} - P$

Plot a segment of the time series of $\Pi^f - P$ and $S^{pe} - P$.

Run the script `PfSpe_timser.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

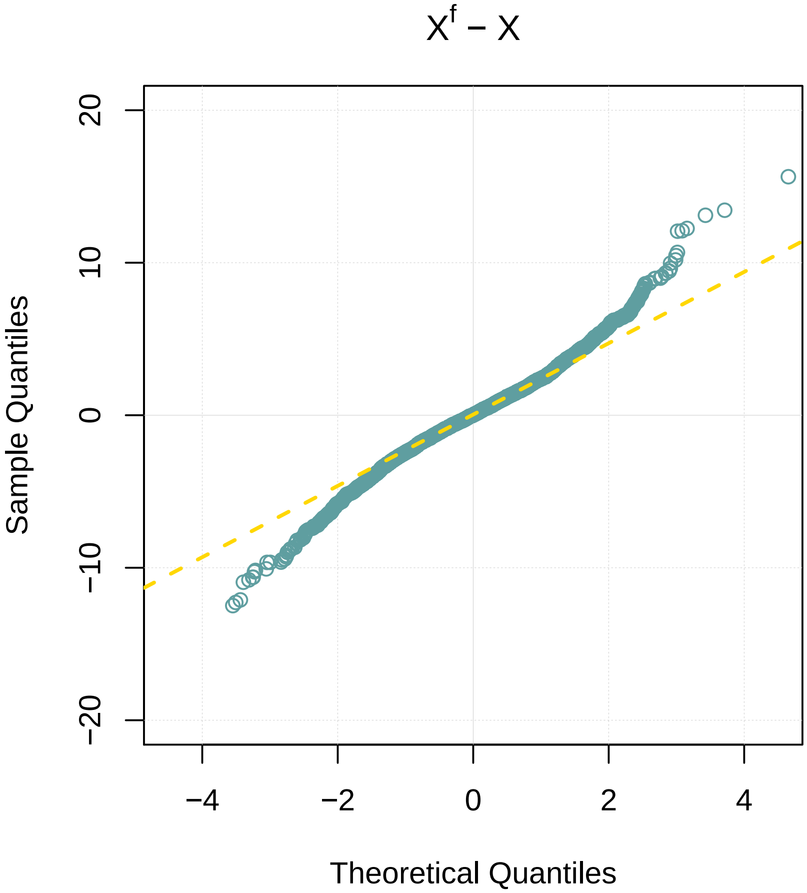

Figure 19: QQ plot for $p(x - m^f)$

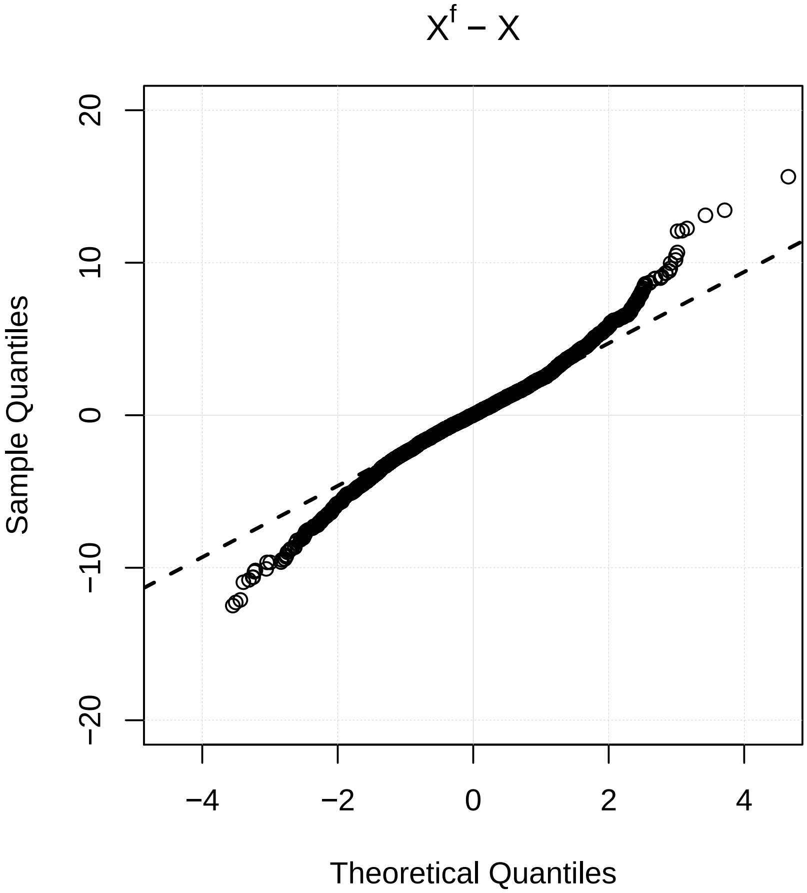

Figure 20: QQ plot for $p(x - m^f)$

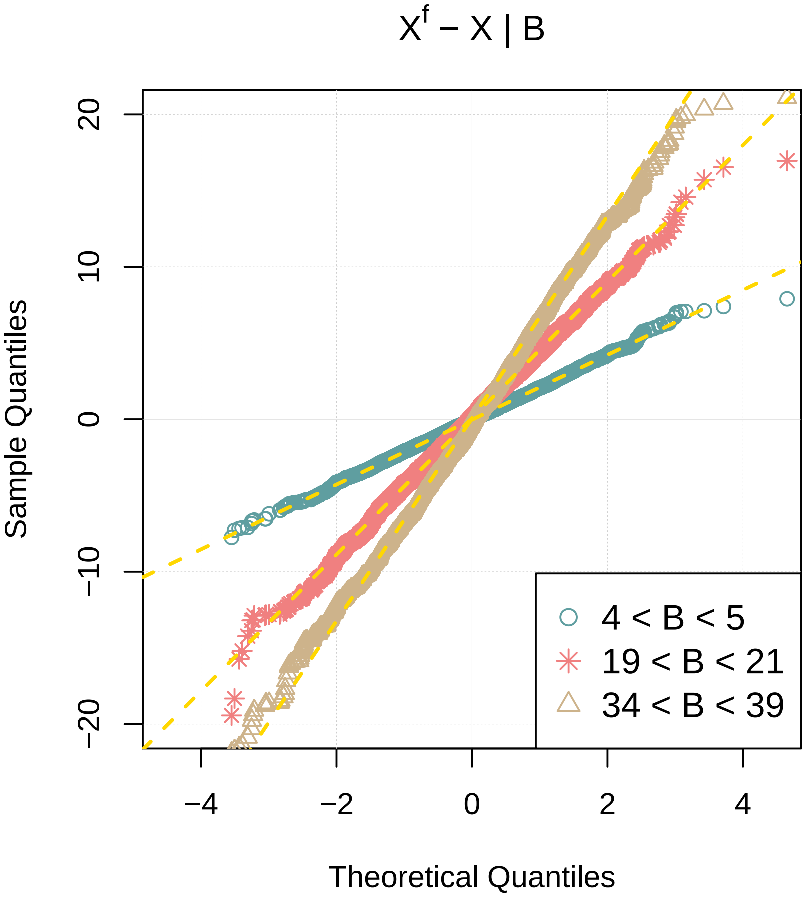

Figure 21: QQ plots for $p(x - m^f | B)$

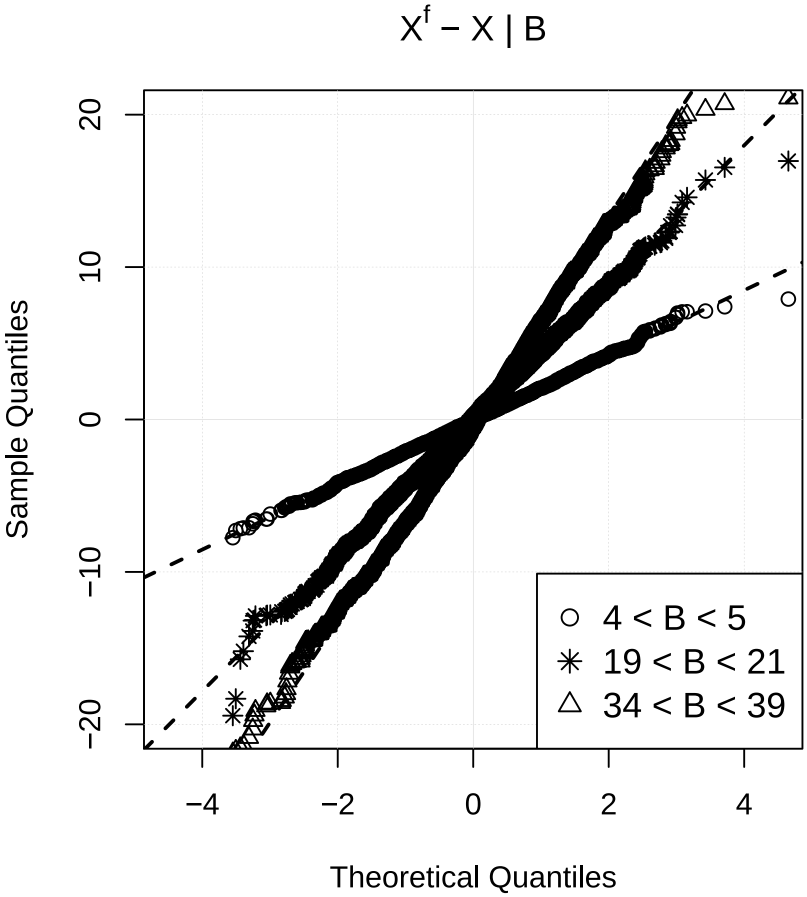

Figure 22: QQ plots for $p(x - m^f|B)$

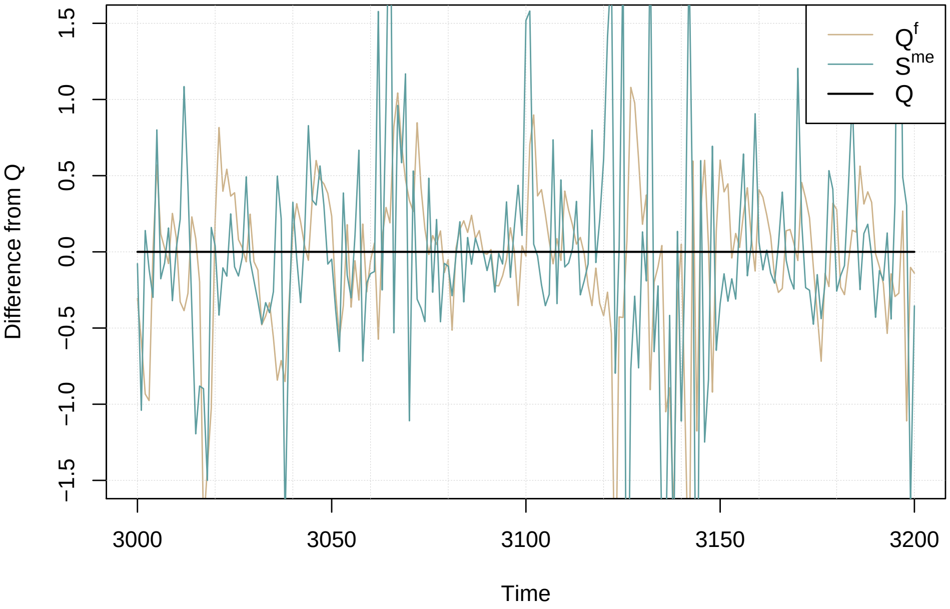

Figure 23: A segment of the time series of $Q^f - Q$ and $S^{me} - Q$.

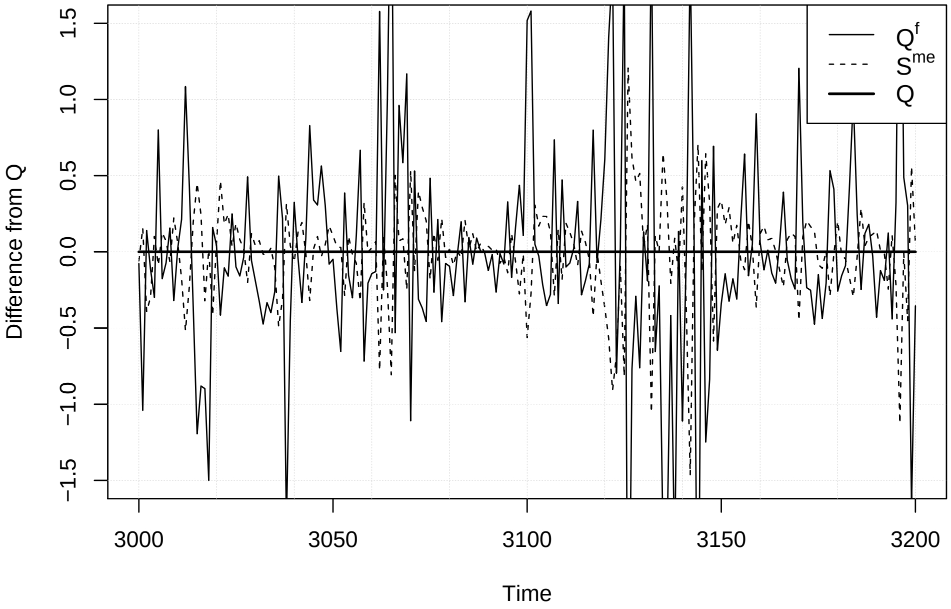

Figure 24: A segment of the time series of $Q^f - Q$ and $S^{me} - Q$.

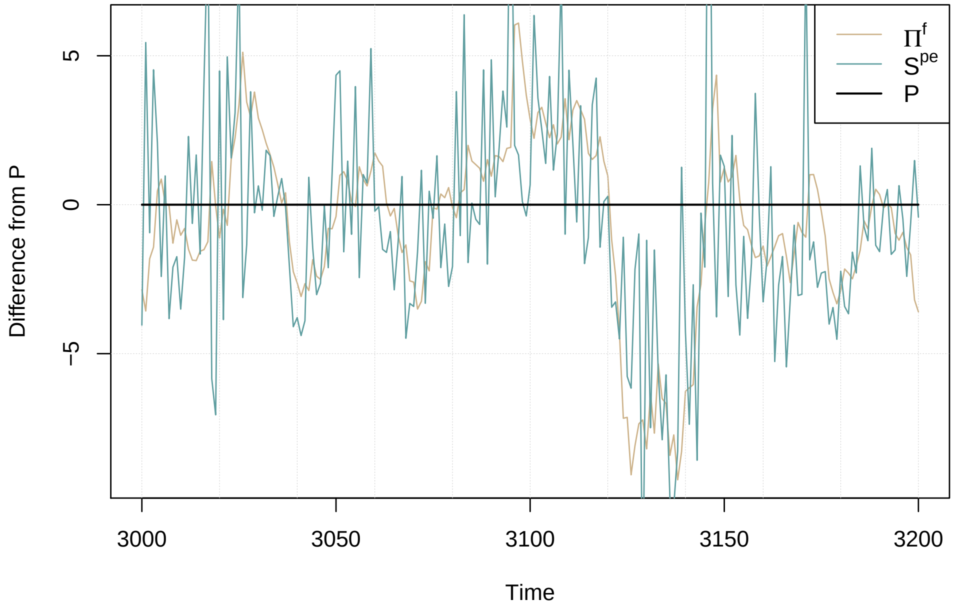

Figure 25: A segment of the time series of $\Pi^f - P$ and $S^{pe} - P$.

Before running the script `PfSpe_timser.R`, you may change the sample size parameters: `parameters$time`, `parameters$L`, and the segment of the time series `t1, t2` such that $1 <$ `t1` $<$ `t2`$<$ `parameters$time`.

The output should be as in Figs.25 and 26 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.5(right).

5.17. Computing and plotting Fig.6(left)

Compute and plot the dependencies of $RMS(bias(Q^f - Q))$ and $RMS(bias(S^{me} - Q))$ on the number of independent assimilation runs L. Here, RMS is defined to be computed over time whereas *bias* over L independent assimilation runs.

Run the script `QfSme_RMSbias.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `QfSme_RMSbias.R`, you may change the sample

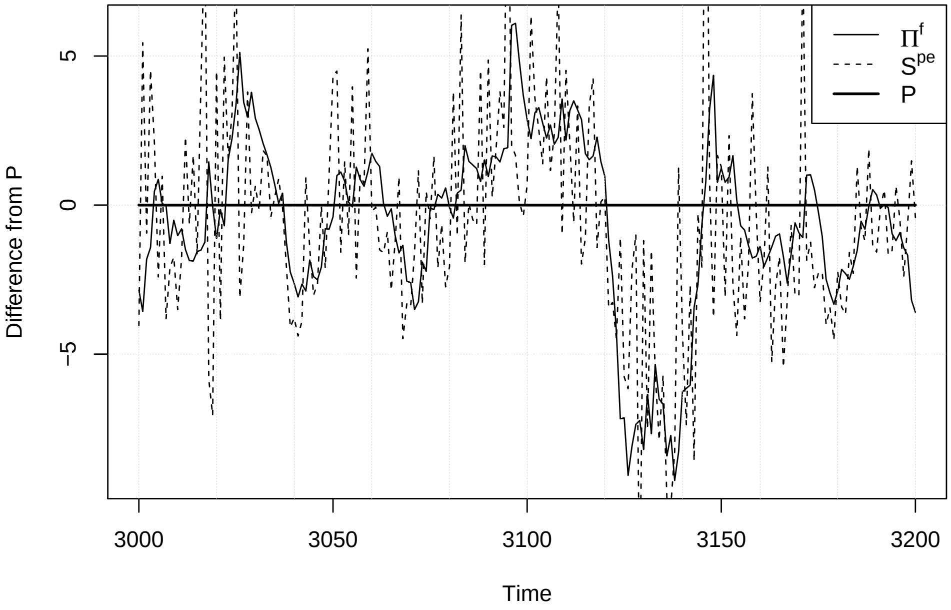

Figure 26: A segment of the time series of $\Pi^f - P$ and $S^{pe} - P$.

size parameters: `parameters$time` and `parameters$L`, and the segment of the time series `t1, t2` such that $1 <$ `t1` $<$ `t2`$<$ `parameters$time`.

The output should be as in Figs.27 and 28 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.6(left).

5.18. Computing and plotting Fig.6(right)

Compute and plot the dependencies of $RMS(bias(\Pi^f - P))$, $RMS(bias(S^{pe} - P))$, $RMS(bias(\Pi^f - \Pi))$, and $RMS(bias(S^{pe} - \Pi))$ on the number of independent assimilation runs L. Here, RMS is defined to be computed over time whereas *bias* over L independent assimilation runs.

Run the script `PfSpe_RMSbias.R`. Note that before running the script, you need to execute the script `Calculate_data_for_B_evaluation.R`.

Before running the script `PfSpe_RMSbias.R`, you may change the sample size parameters: `parameters$time` and `parameters$L`, and the segment of the time series `t1, t2` such that $1 <$ `t1` $<$ `t2`$<$ `parameters$time`.

The output should be as in Figs.29 and 30 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.6(right).

5.19. Compute and plot RMSEs for the state x for misspecified model-error variance Q

Compute and plot the RMSEs for the state x as functions of the coefficient of distortion of Q—for KF, Var, EnKF, HEnKF, and three flavors of HBEF:

1. HBEF with the non-approximated posterior and the Monte-Carlo size $M = 500$.
2. HBEF with the approximated posterior (Inverse Wishart pdfs for the posterior distributions of P and Q).
3. HBEF with no feedback from observations to the covariances at all ($L_o = const$, the posterior is defined to be the “sub-posterior” here).

Run the script `RMSE_Q_distort.R`. Before running the script, you may change the values of n_{time} (`parameters$time` in the script, but not exceeding the parameter `time` set up in `functions.R`), the ensemble size N (`parameters$N` in the script), and the observation-error standard deviation $\sqrt{R}$ (`parameters$std_eta` in the script), for which the computations are to be performed.

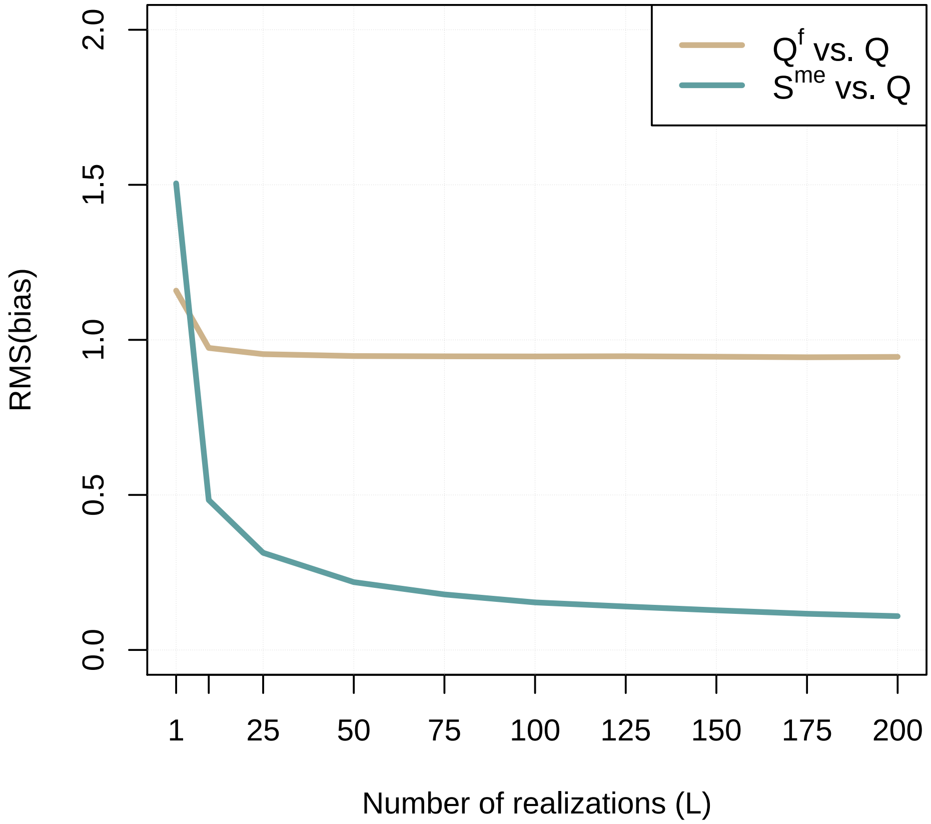

Figure 27: $RMS(bias(Q^f - Q))$ and $RMS(bias(S^{me} - Q))$. Note that RMS is computed over time, whereas bias is computed over L realizations.

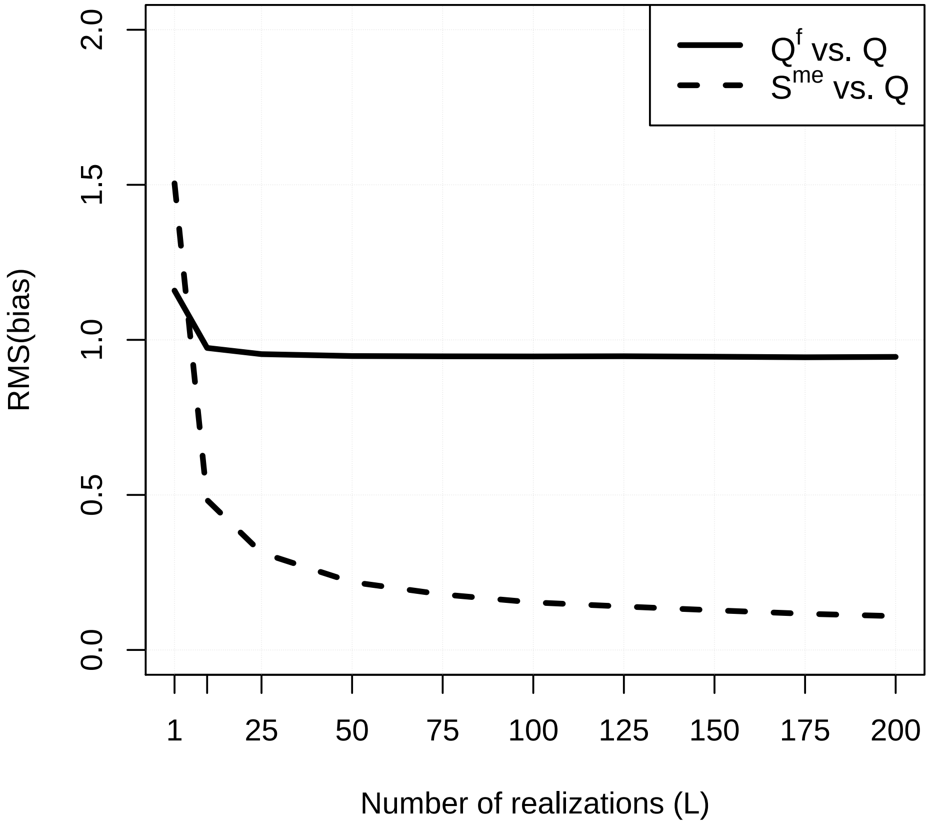

Figure 28: $RMS(bias(Q^f - Q))$ and $RMS(bias(S^{me} - Q))$.Note that RMS is computed over time, whereas bias is computed over L realizations.

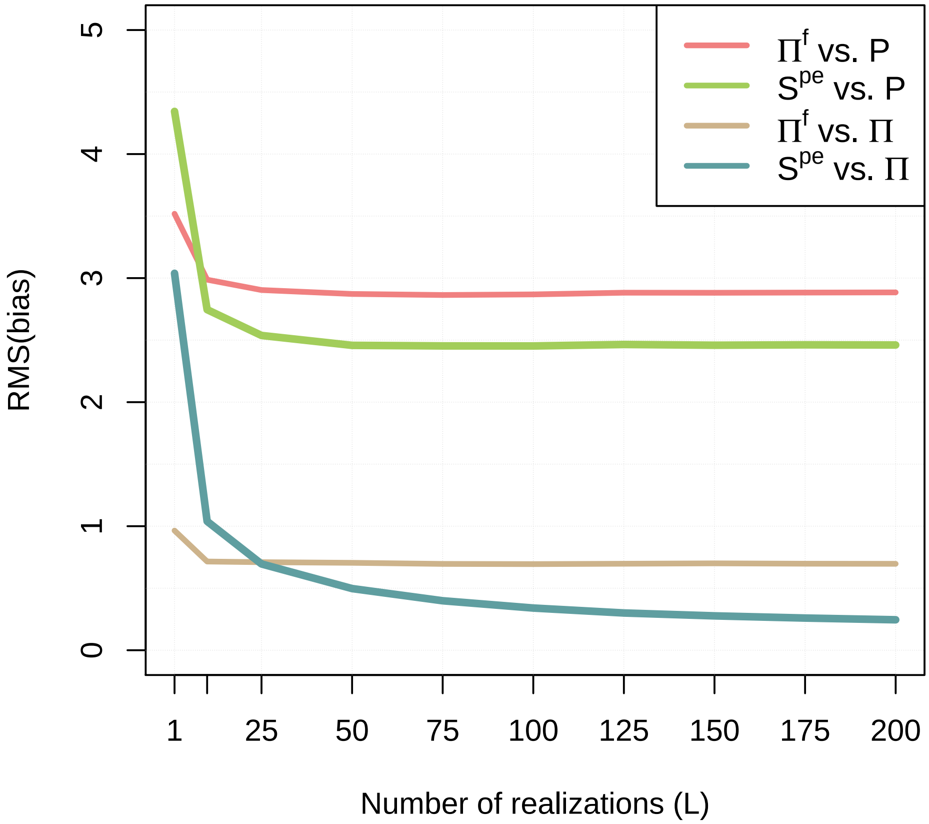

Figure 29: $RMS(bias(\Pi^f - P))$, $RMS(bias(S^{pe} - P))$, $RMS(bias(\Pi^f - \Pi))$, and $RMS(bias(S^{pe} - \Pi))$

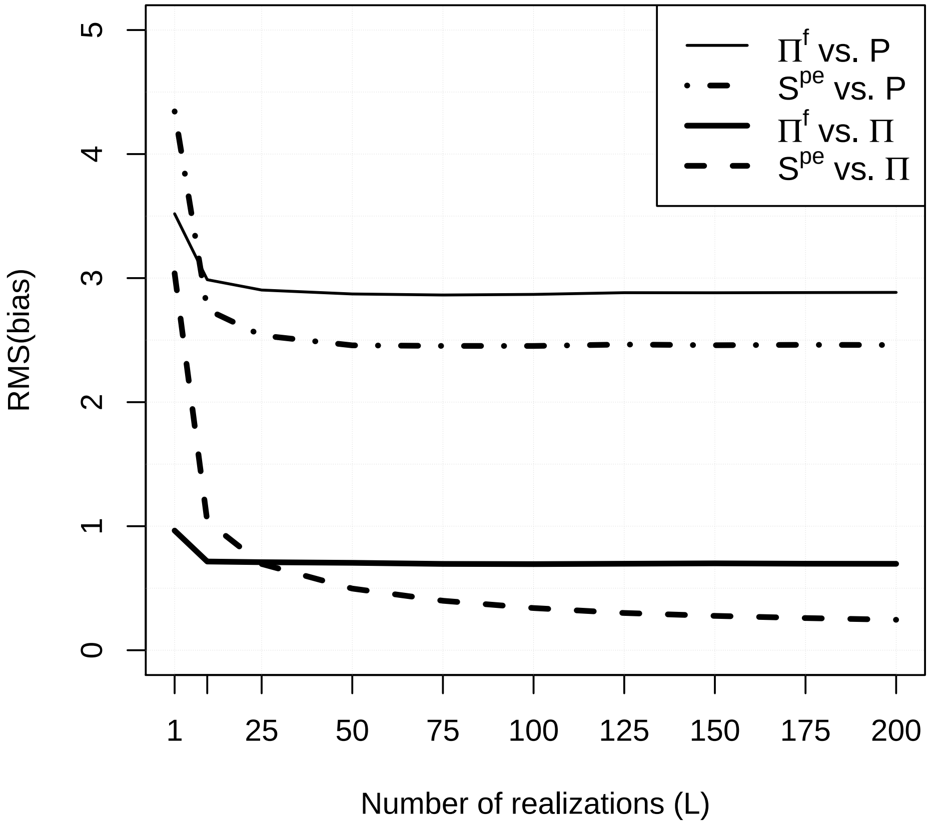

Figure 30: $RMS(bias(\Pi^f - P))$, $RMS(bias(S^{pe} - P))$, $RMS(bias(\Pi^f - \Pi))$, and $RMS(bias(S^{pe} - \Pi))$

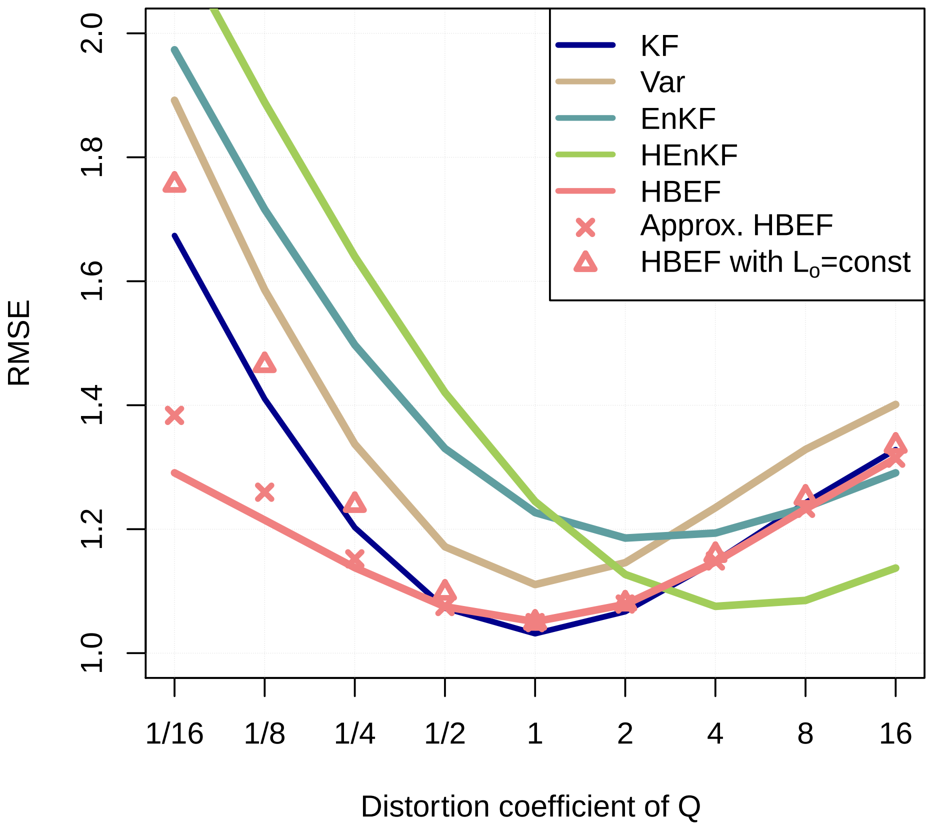

Figure 31: Analysis-error RMSEs as functions of $q_{distort}$ for the 5 filters (KF, Var, EnKF, HEnKF, and HBEF) with distorted $Q^{mod} = Q \cdot q_{distort}$. The crosses denote the HBEF with the approximate posterior

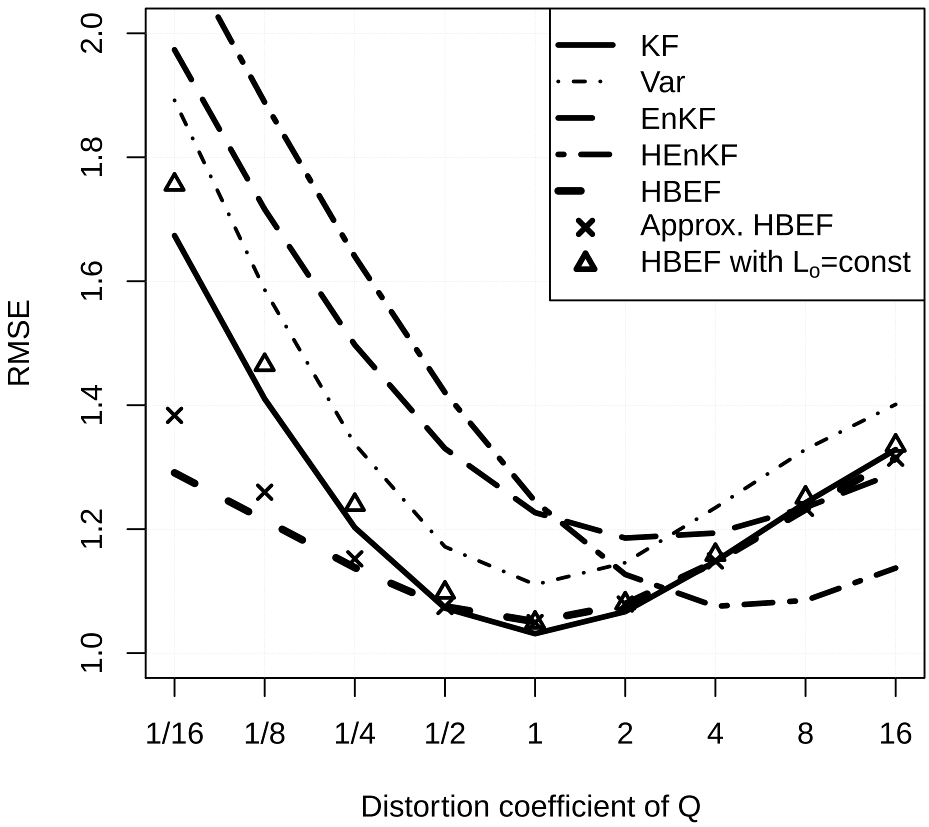

Figure 32: Analysis-error RMSEs as functions of $q_{distort}$ for the 5 filters (KF, Var, EnKF, HEnKF, and HBEF) with distorted $Q^{mod} = Q \cdot q_{distort}$. The crosses denote the HBEF with the approximate posterior

The output should be as in Figs.31 and 32 in this supplementary text (in color and in black-and-white).

In the paper, this is Fig.9.