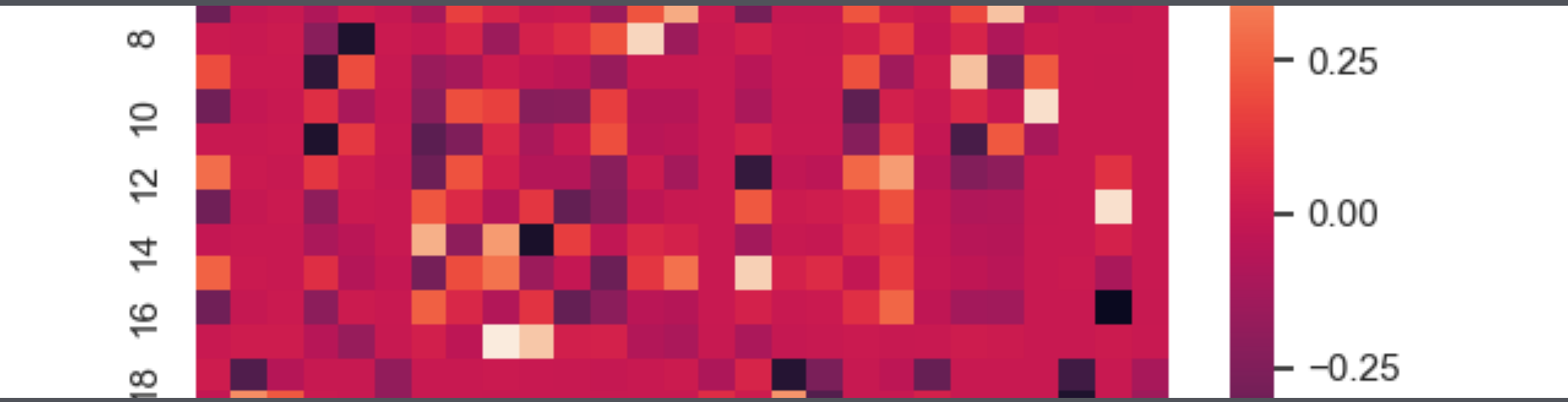
michael.taylor@reading.ac.uk
patternizer.github.io

University of Reading

L1C ENSEMBLE FCDR



Progress update

HARMONISE → ENSEMBLE

- **9 harmonised AVHRR Easy FCDRa** (MetOp-A → NOAA-11)
- **3 IR channels** (3.7 µm, 11 & 12 µm)
- 3 (or 4) **harmonisation coefficients** (without WV) **per channel per sensor** → 27 (or 36) **coefficients per channel**
- **We know the uncertainty on each coefficient**
- **We know their correlation matrices [27 x 27] or [36 x 36]**

'best-case' **harmonised** AVHRR Easy FCDR

Ch3B (3.7 µm)

$$L = a0 + ((Lict * (0.985140 + a1)) / (Cict - Cs)) * (Ce - Cs) + a2 * Tinst + a3 * f(WV)$$
$$L = a0 + ((Lict * (0.985140 + a1)) / (Cict - Cs) + a2 * (Ce - Cict)) * (Ce - Cs) + a3 * Tinst + a4 * f(WV)$$

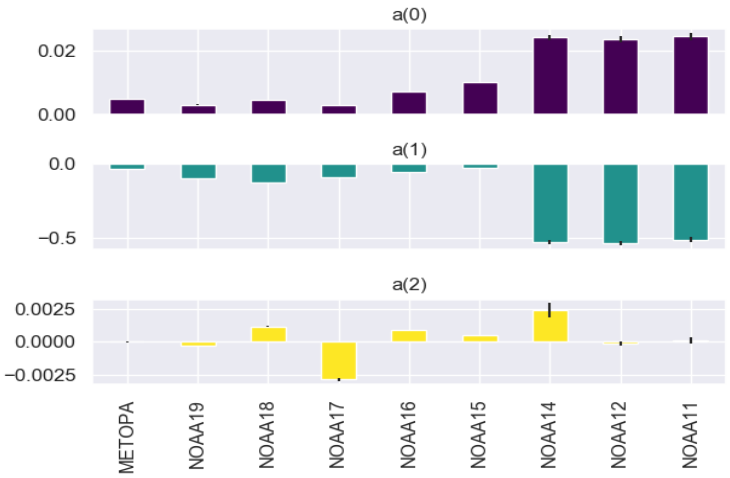
Ch4 (11 µm) & Ch5 (12 µm)

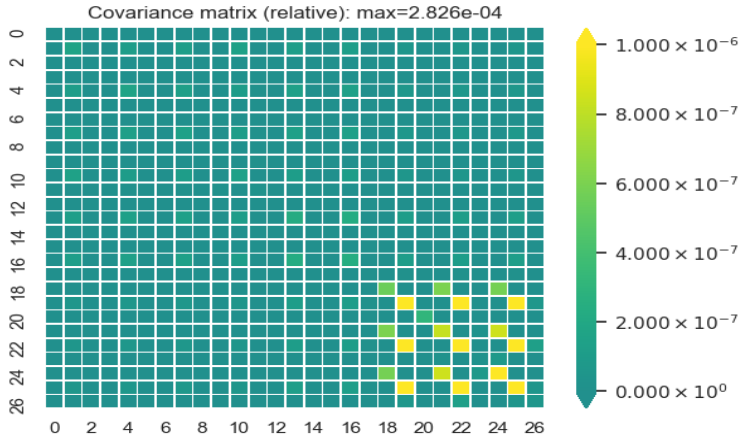
Q: Can we generate another 10 FCDRs within the range of uncertainty but having a similar inter-sensor correlation structure to the best-case?
Q: Why?

THE BEST CASE

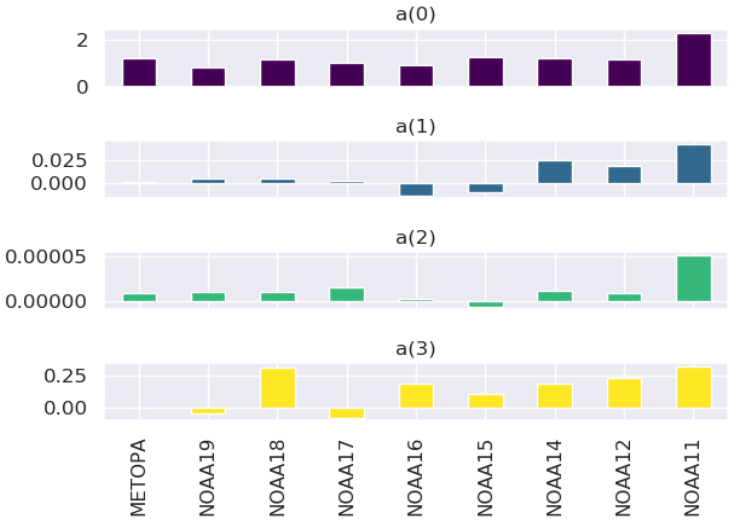
University of Reading

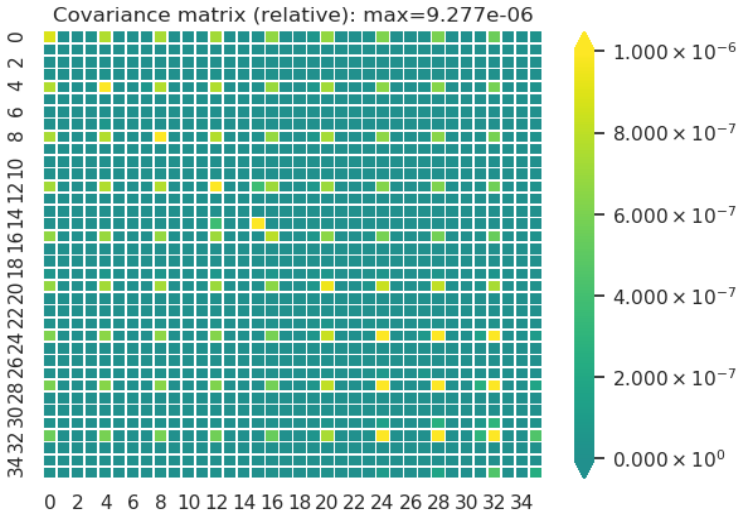
3.7 µm





11 µm

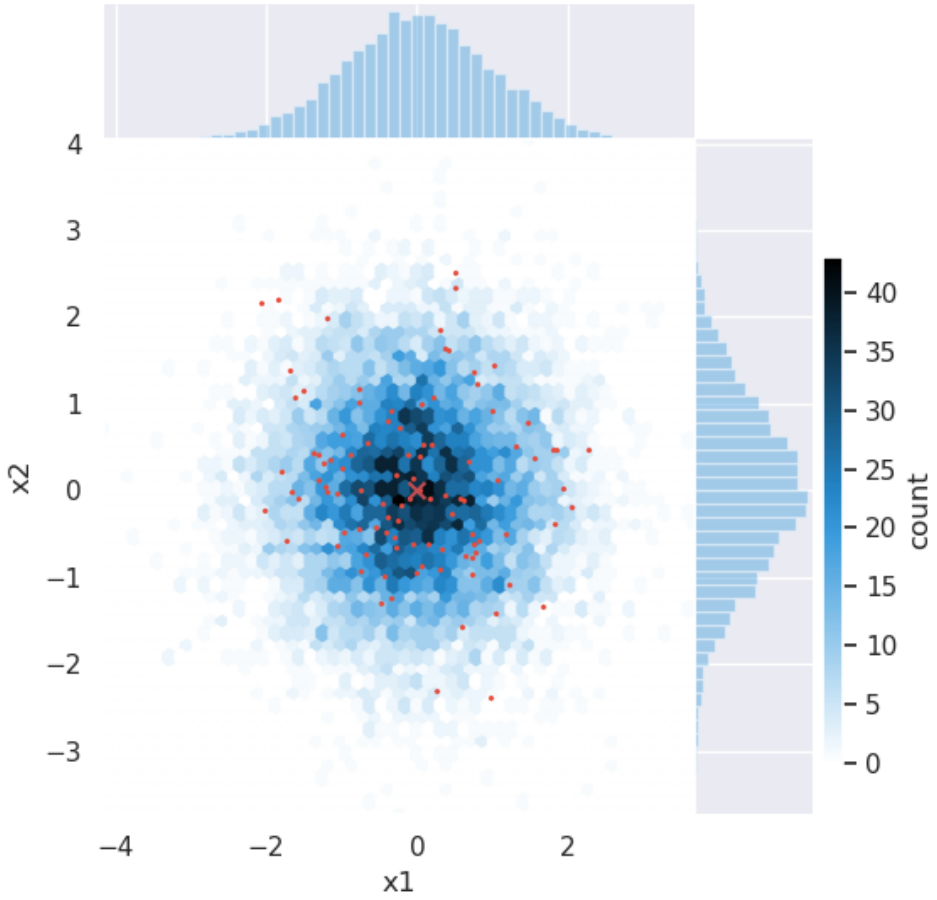




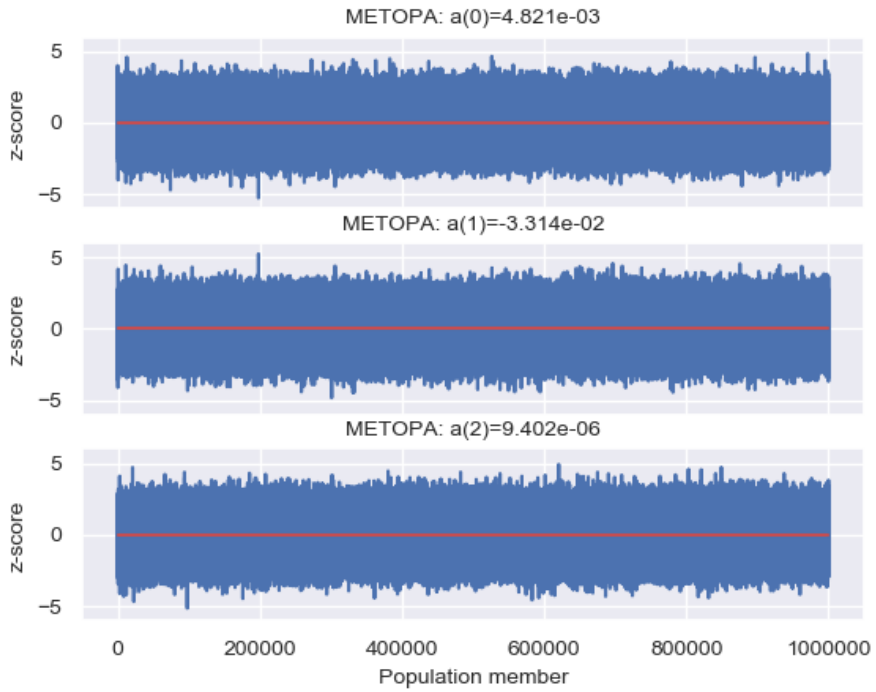
MONTE CARLO: 2D CHECK

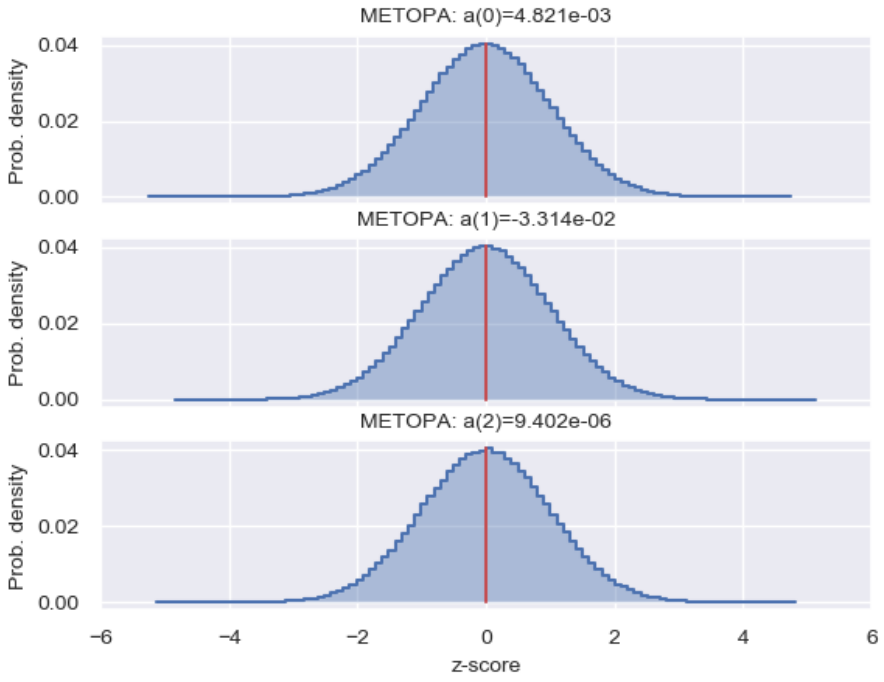
Blue = 10000 draws from bi-normal X~N(μ=[0,0], σ=[1,1])

Red = 100 random draws with numpy.random.multinormal_normal(mean(X), cov(X), 100))

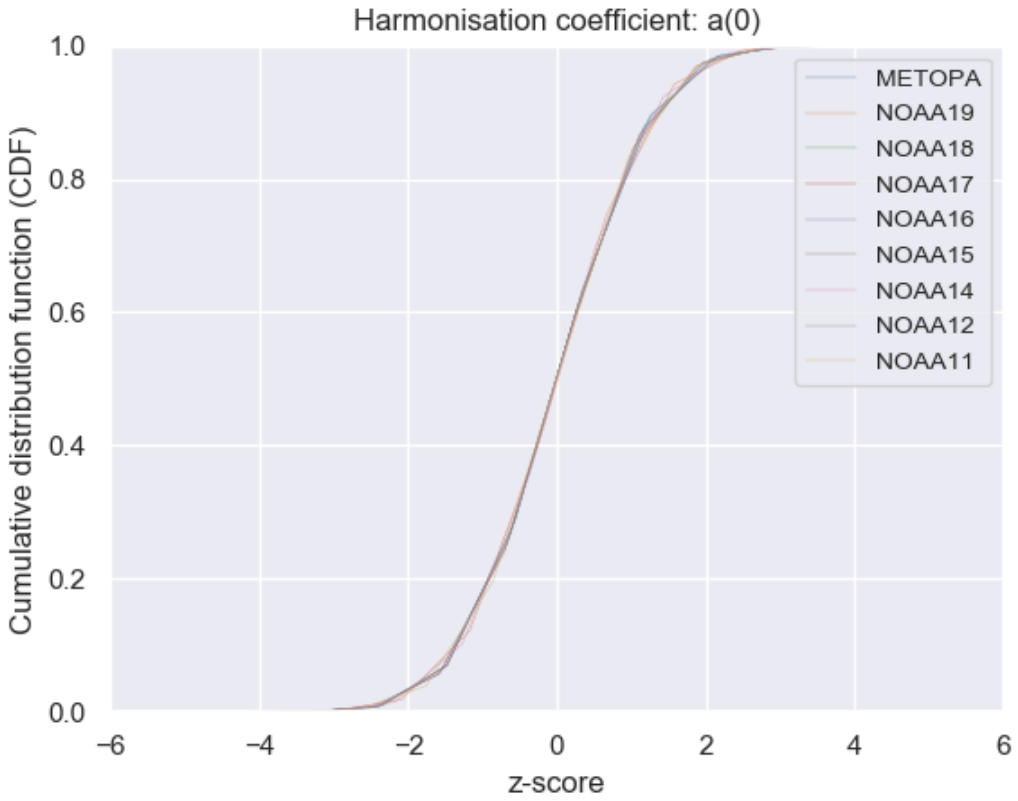


MONTE CARLO: 27D (& 36D)





METHOD #1: CDF NORM

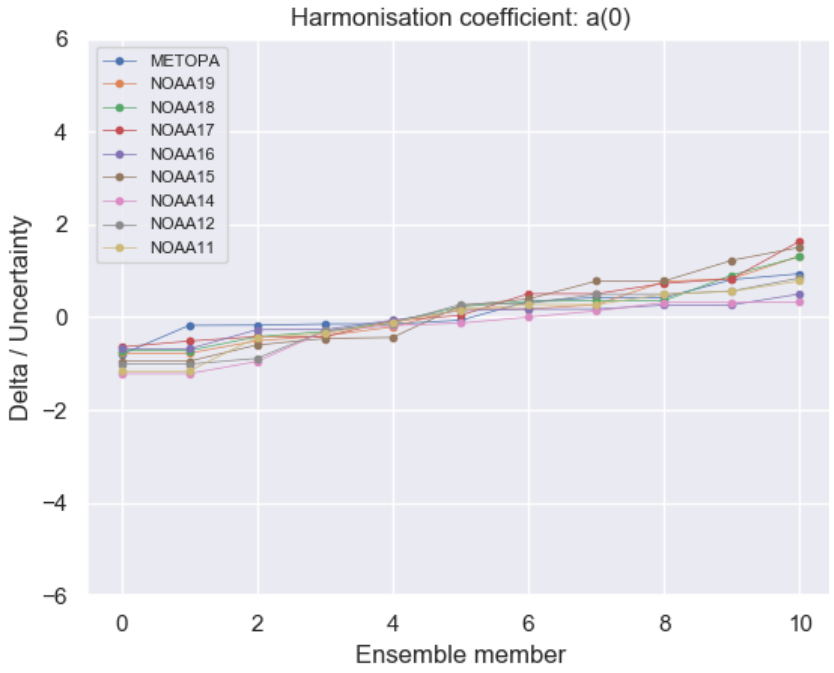


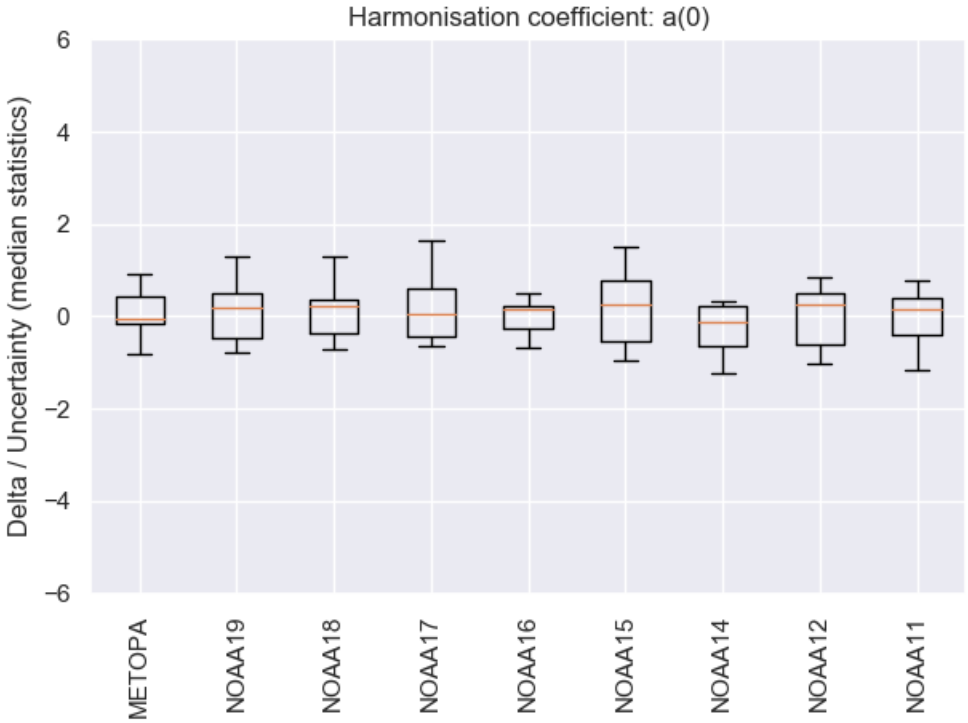
Use sorted draw CDFs to find deciles (& store indices)

Loop over all draw vectors & calculate norm for each decile

Index [min(Norm)] → ensemble

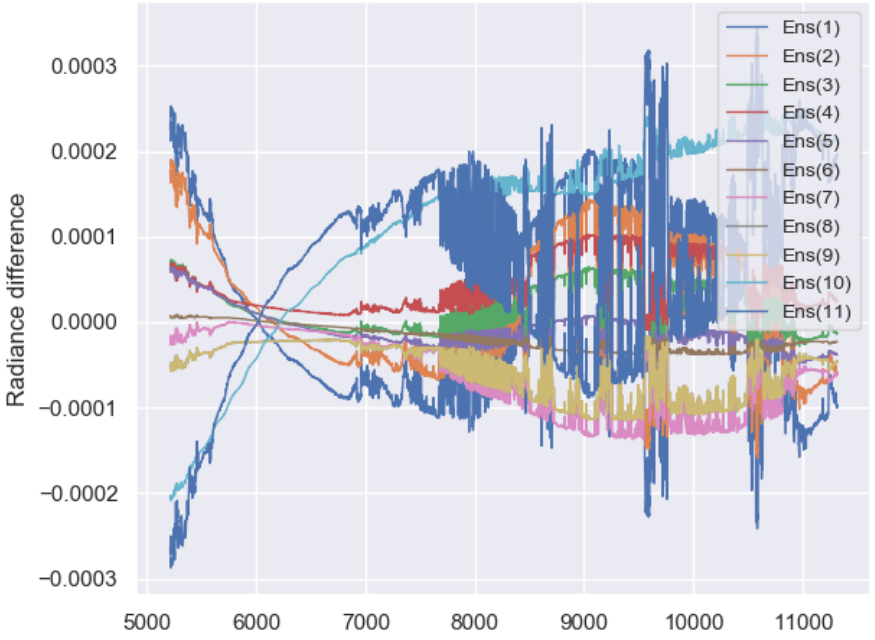
ENSEMBLE STATS

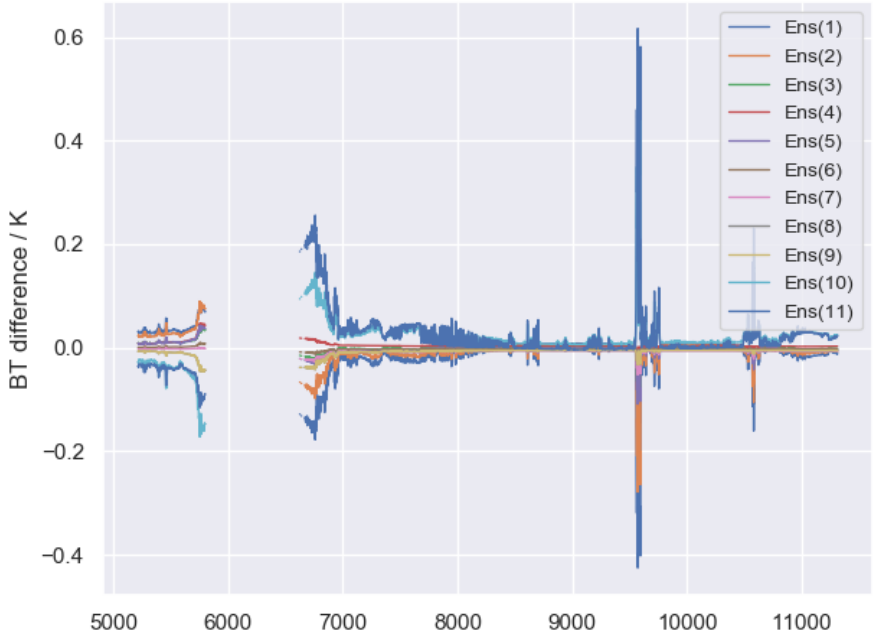




We would like the sorted ensemble to pass through (5,0) and present a CDF-shaped spread – it more or less does

ENSEMBLE RADIANCE

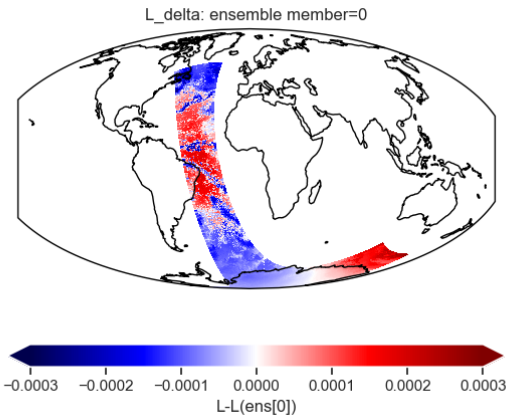


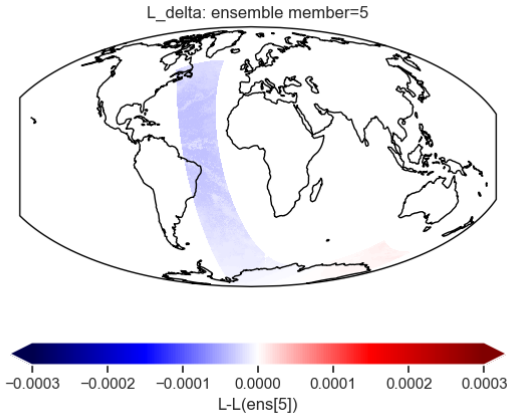


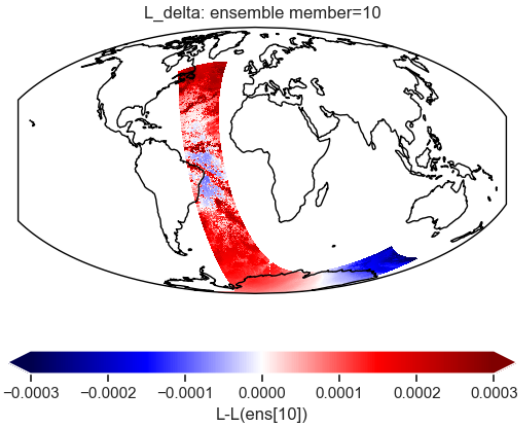
Using L1B counts and temperatures from an orbit from MetOp-A, I calculated the radiance for each ensemble member and then used LUTs from the L1C Easy FCDR (thanks James!) to convert to BTs:

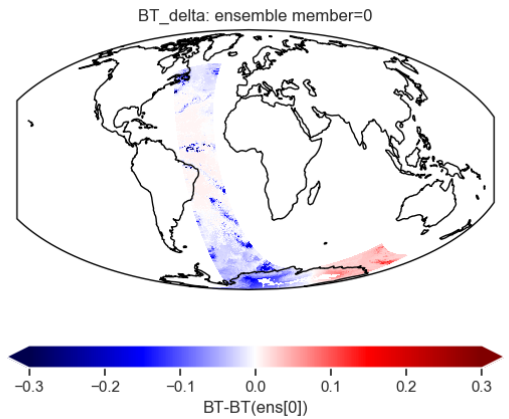
https://github.com/FIDUCEO/MMD_HARM

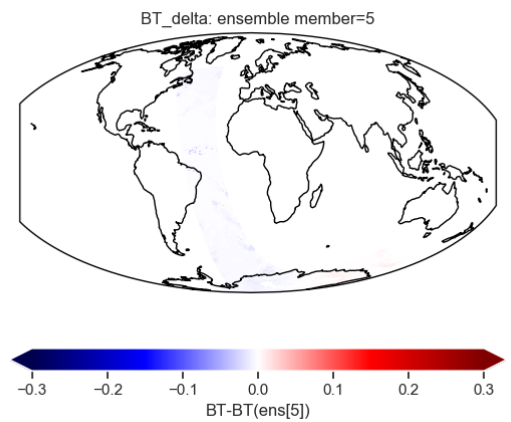
ORBITAL ENSEMBLE DELTAS

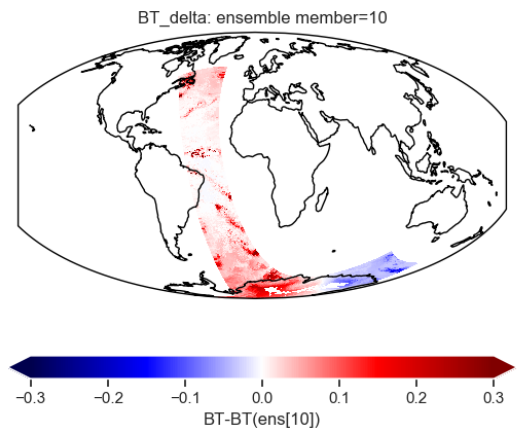












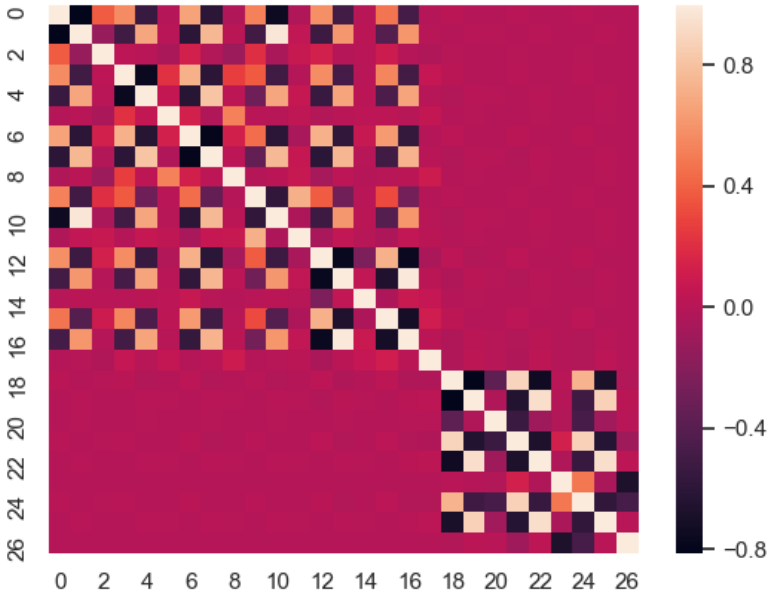
METHOD #2: PCA-MC CHECK

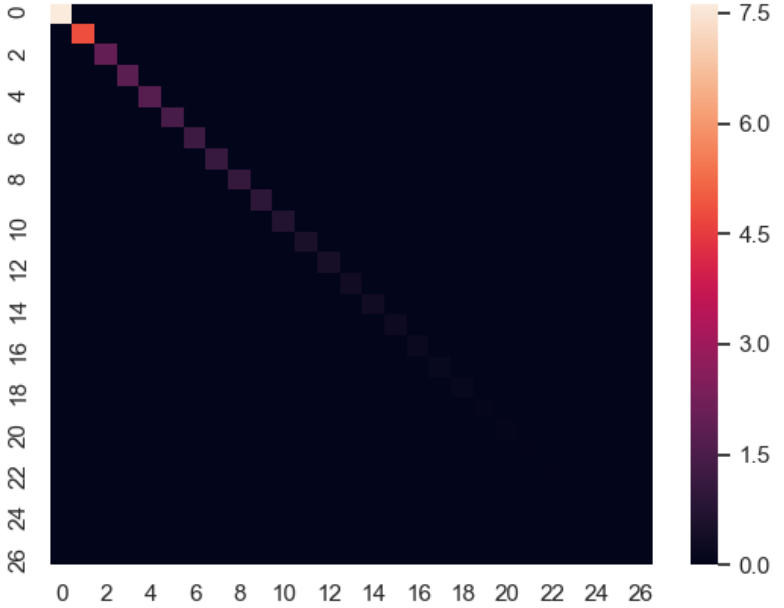
Suppose there are n-samples of p-variables → [n × p] matrix

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

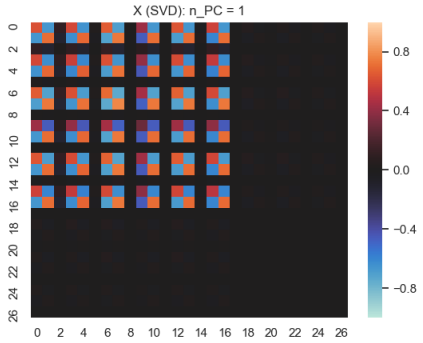
$$\begin{cases} z_1 = l_{11}x_1 + l_{12}x_2 + \cdots + l_{1p}x_p \\ z_2 = l_{21}x_1 + l_{22}x_2 + \cdots + l_{2p}x_p \\ \cdots\cdots\cdots\cdots \\ z_m = l_{m1}x_1 + l_{m2}x_2 + \cdots + l_{mp}x_p \end{cases}$$

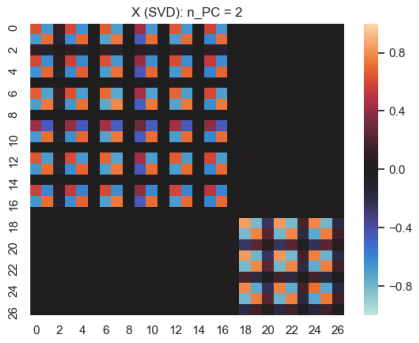
$$l_{i1}^2 + \cdots + l_{ip}^2 = 1$$

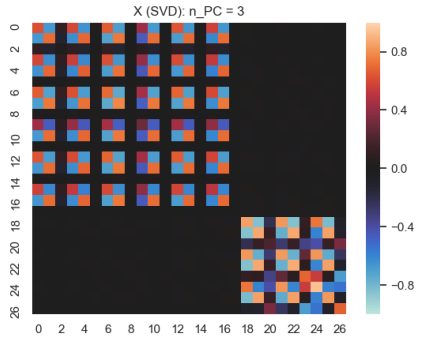


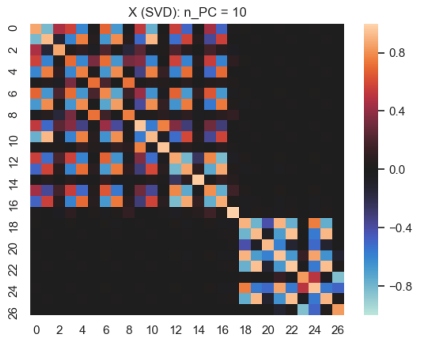


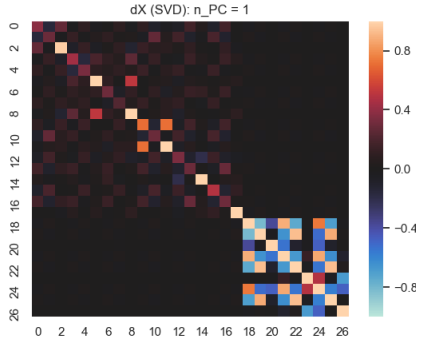
PCA

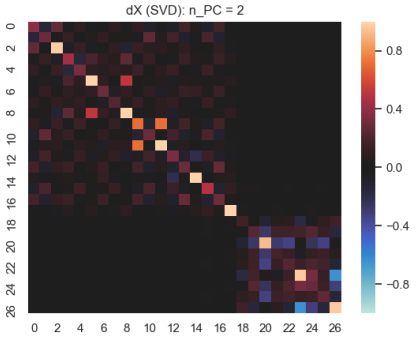

X (SVD): n_PC = 1

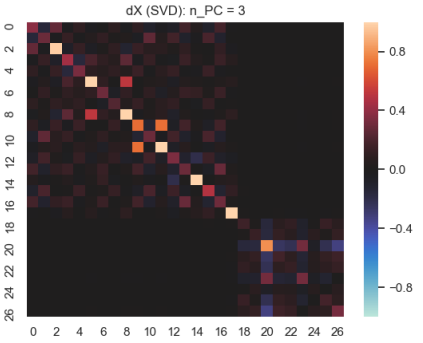

X (SVD): n_PC = 2

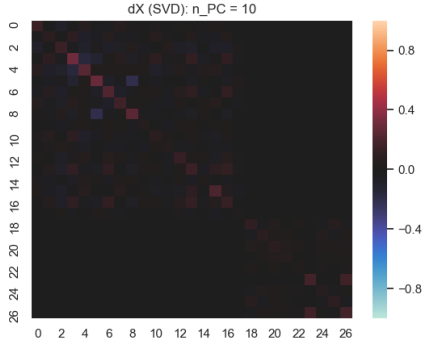

X (SVD): n_PC = 3


X (SVD): n_PC = 10


dX (SVD): n_PC = 1


dX (SVD): n_PC = 2


dX (SVD): n_PC = 3


dX (SVD): n_PC = 10

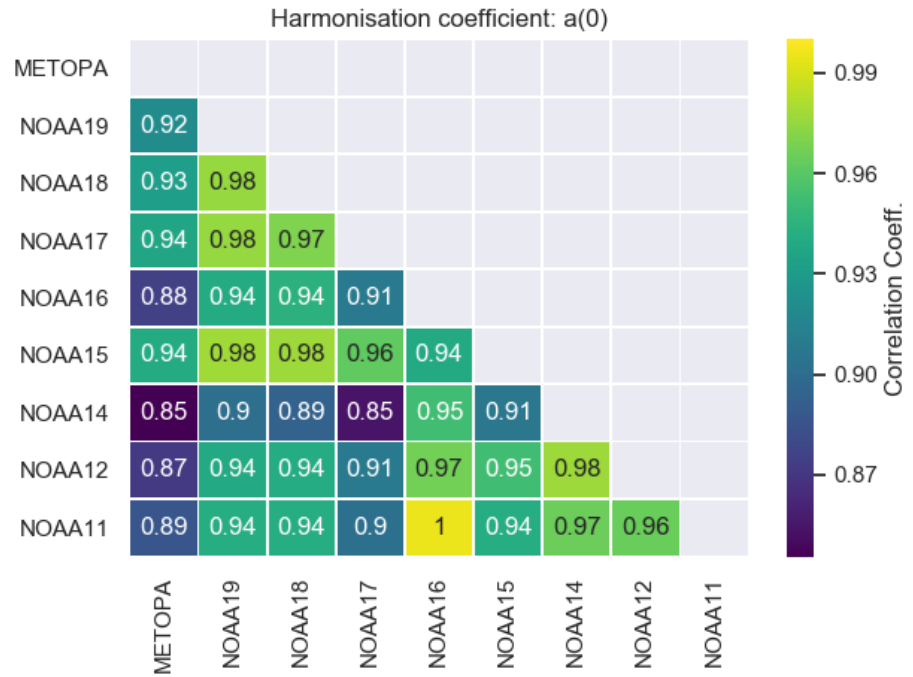
University of Reading

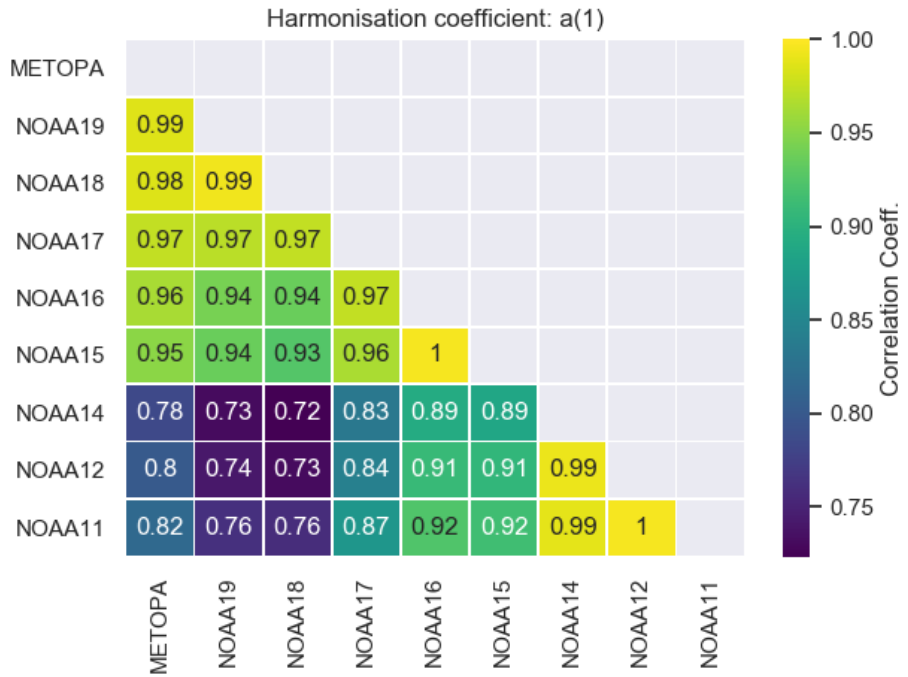
THANKS!

I've put the python code here for re-use / revision:
https://github.com/patternizer/ENSEMBLE_SST

University of Reading

ENSEMBLE CORRELATIONS





MC ON EIGENVECTORS