

Report and notes - ICBM recalibration

Lorenzo Menichetti

11/24/2020

Data pretreatment

The initial data are read from the file “input_data_0_36.xlsx”, in this folder. Climatic data are from another calculation, outside this repository.

```
yields_Ultuna<-read_excel("./input_data_0_36.xlsx", sheet=1)
re_Ultuna<-read.csv("../re_clim/Ultuna_FRAME56_calculated_re.csv")[,2:16]
```

Afterwards I create a bulk density time series by interpolating values from 1956 and 2017, with one value per each treatment, and then extend this to one value for each plot by repeating the average value.

```
Ultuna_BD_start=1.44
Ultuna_BD_end_list=c(1.43,1.40,1.28,1.21,1.25,1.38,1.21,1.34,1.12,1.24,1.20,1.28,1.05,1.023,1.02)
```

The dimensions of these objects are the same that we will use for most other data treatment objects, either 15 x 62 (treatment level) or 60 x 62 (plot level)

```
dim(Ultuna_BD_timeseries)
```

```
## [1] 15 62
dim(Ultuna_BD_timeseries_long)
```

```
## [1] 60 62
```

Year 1976, missing, is then filled with an average of all other years for each treatment. I then create the time series objects, according to this time series:

```
Ultuna_date_vector_year
```

```
## [1] "1956-01-01" "1957-01-01" "1958-01-01" "1959-01-01" "1960-01-01"
## [6] "1961-01-01" "1962-01-01" "1963-01-01" "1964-01-01" "1965-01-01"
## [11] "1966-01-01" "1967-01-01" "1968-01-01" "1969-01-01" "1970-01-01"
## [16] "1971-01-01" "1972-01-01" "1973-01-01" "1974-01-01" "1975-01-01"
## [21] "1976-01-01" "1977-01-01" "1978-01-01" "1979-01-01" "1980-01-01"
## [26] "1981-01-01" "1982-01-01" "1983-01-01" "1984-01-01" "1985-01-01"
## [31] "1986-01-01" "1987-01-01" "1988-01-01" "1989-01-01" "1990-01-01"
## [36] "1991-01-01" "1992-01-01" "1993-01-01" "1994-01-01" "1995-01-01"
## [41] "1996-01-01" "1997-01-01" "1998-01-01" "1999-01-01" "2000-01-01"
## [46] "2001-01-01" "2002-01-01" "2003-01-01" "2004-01-01" "2005-01-01"
## [51] "2006-01-01" "2007-01-01" "2008-01-01" "2009-01-01" "2010-01-01"
## [56] "2011-01-01" "2012-01-01" "2013-01-01" "2014-01-01" "2015-01-01"
## [61] "2016-01-01" "2017-01-01"
```

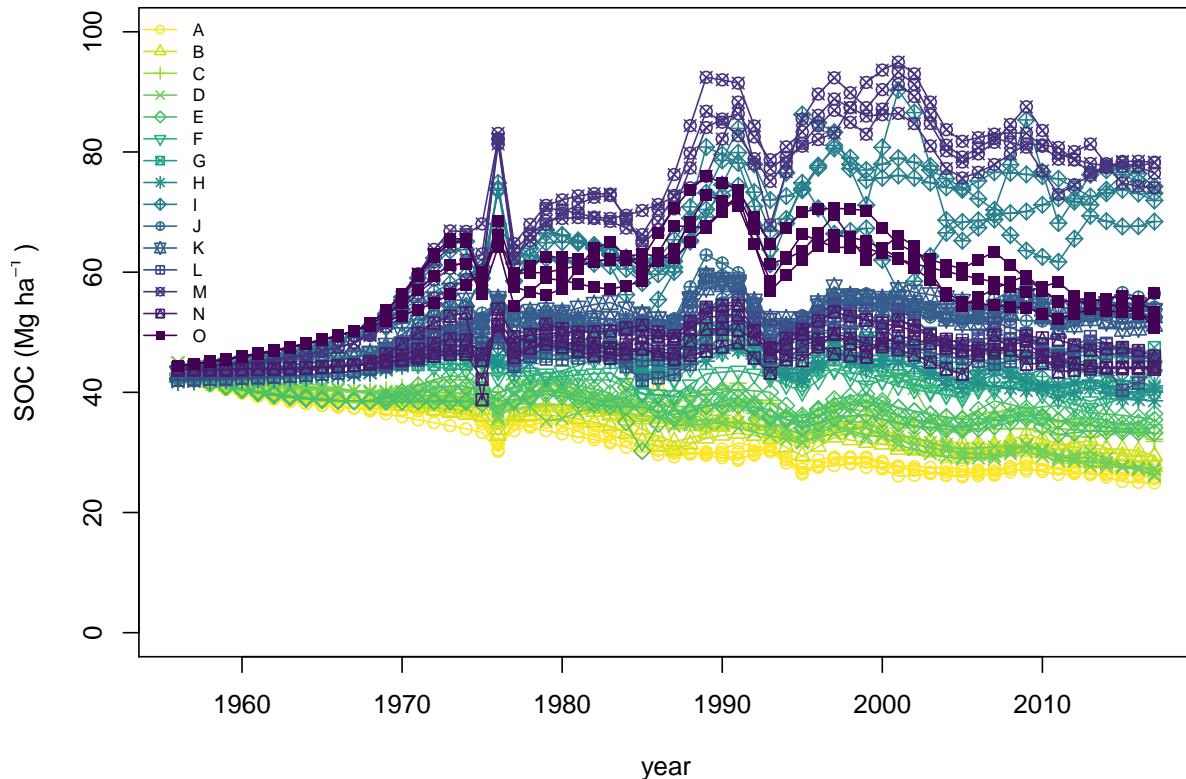
And I fill all the empty years with interpolation, to have one measurement per year. The C% data are then transformed in SOC stocks considering 20 cm depth working with the matrices.

```

depth=20 #depth of soil considered in cm
Ultuna_SOC_timeseries_long<-Ultuna_C_timeseries_long_interp*depth*Ultuna_BD_timeseries_long

```

I end up with the following time series for what concerns SOC:



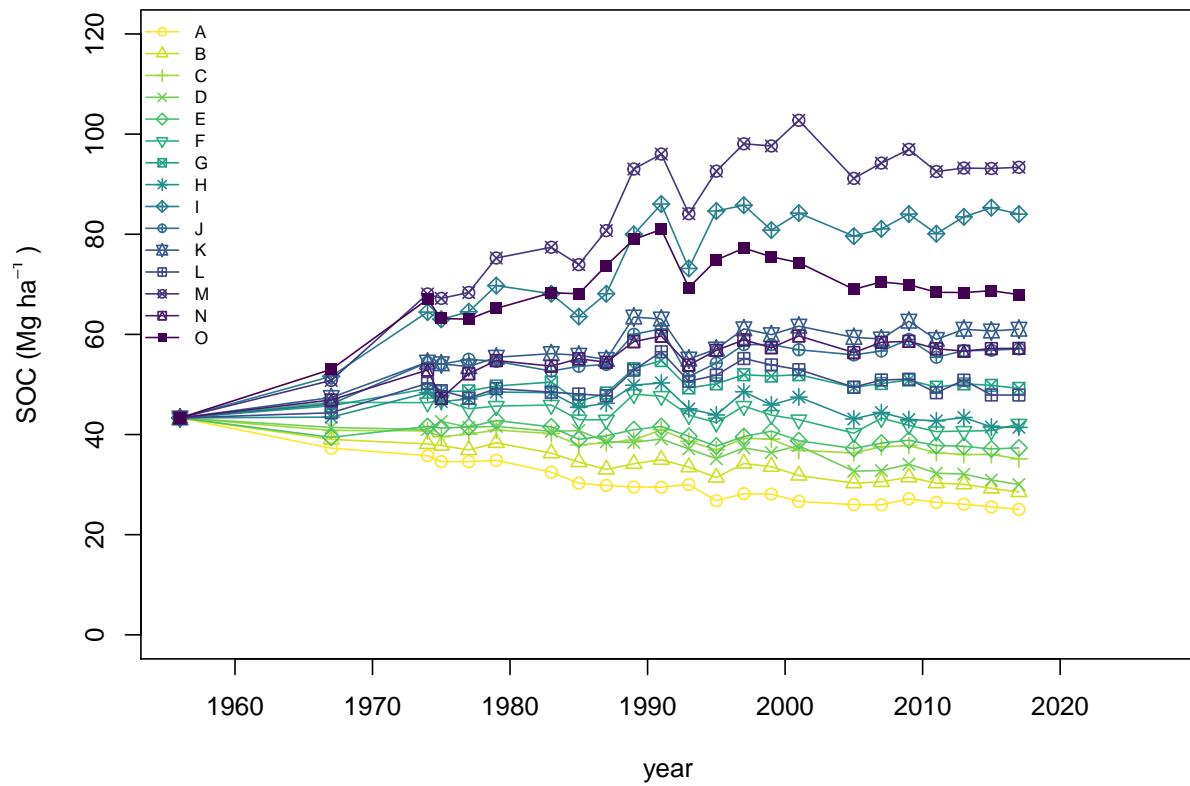
But we decided to use a calculation from Thomas with a different equivalent soil depth value (the file is in this folder, **SOC_equivalent_depth_1956-2019_TK**):

```

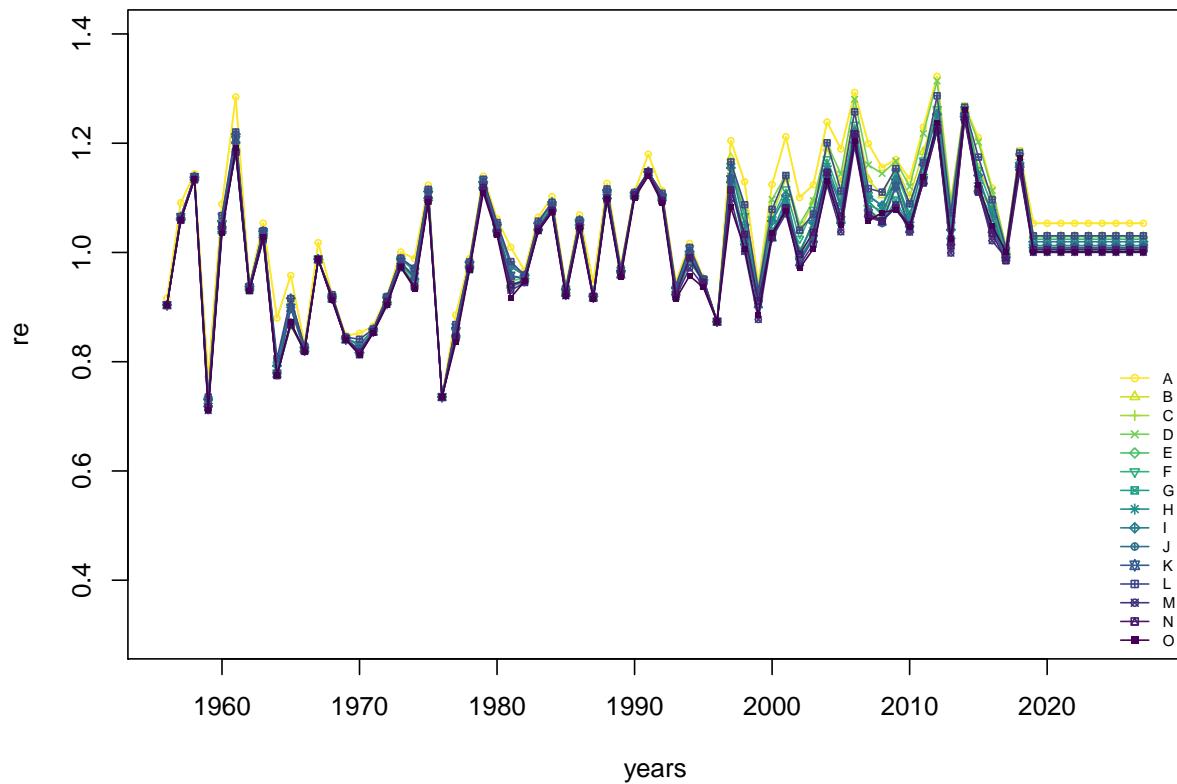
Ultuna_SOC_timeseries_TK <- as.data.frame(read_excel("./SOC_equivalent_depth_1956-2019_TK.xlsx",
sheet = "SOCeq_depth")[, -1]) * 10^-3

```

Which contains the following data, here plotted from the final object that will be the input of the calibration:

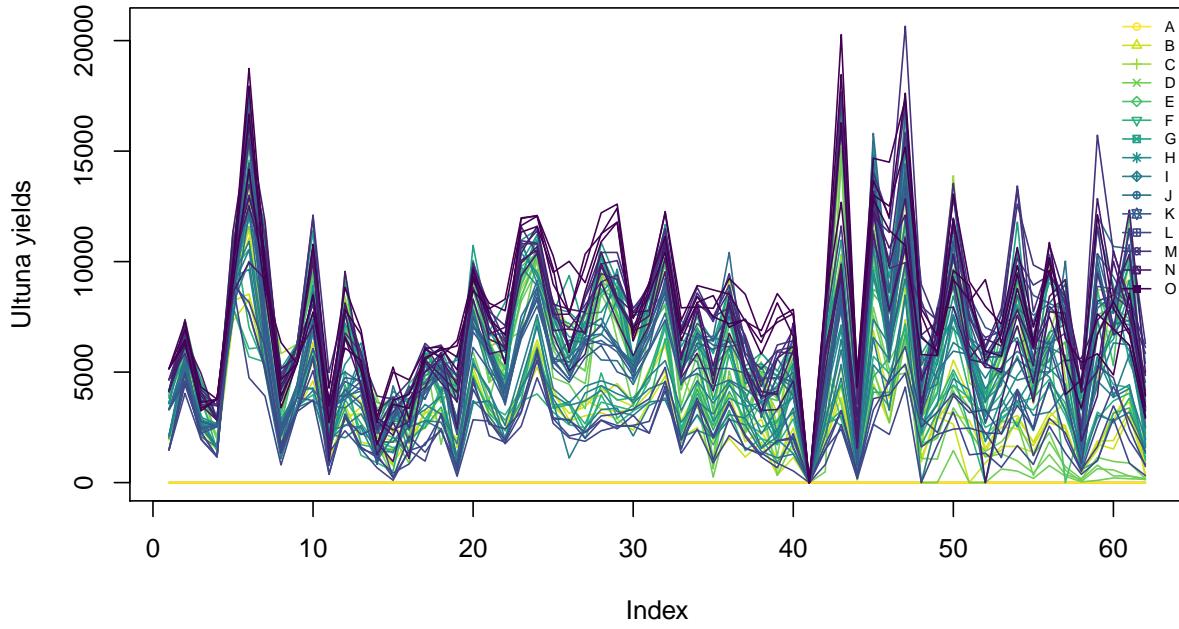


Reclim values are read from a separate file, calculated with my package, and are in a matrix 15 x 62 (treatment level) that is extended to 60 x 62. The following plot is also plotted directly from the object that ends in



Now there's an additional complication to take care of: since JAGS, the language I used this time for the Bayesian calibration, does not accept IF loops I have to create a series of input tables if the same 60 x 62 dimensions for each input class. Each table must then report zero for all treatments not receiving that specific input. The Bayesian calibration then runs for each single plot a model that sums all the possible Y pools, one for each h value, but when the input is 0 of course the Y pool is also zero.

The actual yields values that will be used after this calculation are the following:



The crops are reclassified by combining several classes into one in order to use the new S:R values of the new model version(4 classes only). I create some selection vectors to produce then one table for each class with 0 where that class is not planted.

```
# create one yields vector for each class of Martin's S:R
cereals_vector <- yields_Ultuna$crop == "fodder" | yields_Ultuna$crop ==
  "winter_small_grains" | yields_Ultuna$crop == "spring_small_grains"
roots_vector <- yields_Ultuna$crop == "root_crop"
oil_vector <- yields_Ultuna$crop == "winter_oil_seeds"
maize_vector <- yields_Ultuna$crop == "maize"
```

(the gate “|” in R means “OR”)

After a lof of rearrangement, I come up with a list of inputs for the JAGS model:

```
names(calib_data_Ultuna)
```

```
## [1] "SOC_Ultuna"                  "SOC_init_Ultuna"
## [3] "error_SOC_Ultuna"           "error_SOC_multiplier_Ultuna"
## [5] "Yields_cereals_Ultuna"      "Yields_root_crops_Ultuna"
## [7] "Yields_oilseeds_Ultuna"     "Yields_maize_Ultuna"
## [9] "I_FYM_Ultuna"                "I_GM_Ultuna"
## [11] "I_PEA_Ultuna"                "I_SAW_Ultuna"
## [13] "I_SLU_Ultuna"                "I_STR_Ultuna"
## [15] "N_Ultuna"                     "re_Ultuna"
## [17] "J_Ultuna"
```

The result is in the same folder cointaining this report, in ODS format, as “**Input_Ultuna.ods**” and “**Input_Lanna.ods**”

Table 1: The input data structure (here for Ultuna)

Sheet name	Description
SOC_Ultuna	SOC, 15 treatments x lenght of simulation years
SOC_init_Ultuna	SOC initialization values, 15 treatments
error_SOC_Ultuna	SOC mean errors, 15 treatments
error_SOC_multiplier_Ultuna	SOC error weight
Yields_cereals_Ultuna	Cereals yields, 15 treatments x lenght of simulation years
Yields_root_crops_Ultuna	Root crops yields, 15 treatments x lenght of simulation years
Yields_oilseeds_Ultuna	Oil seeds yields, 15 treatments x lenght of simulation years
Yields_maize_Ultuna	Maize yields, 15 treatments x lenght of simulation years
I_FYM_Ultuna	Inputs farmyard manure, 15 treatments x lenght of simulation years
I_GM_Ultuna	Inputs green manure, 15 treatments x lenght of simulation years
I_PEA_Ultuna	Inputs peat, 15 treatments x lenght of simulation years
I_SAW_Ultuna	Inputs sawdust, 15 treatments x lenght of simulation years
I_SLU_Ultuna	Inputs sludges, 15 treatments x lenght of simulation years
I_STR_Ultuna	Inputs straw, 15 treatments x lenght of simulation years
N_Ultuna	Lenght of the simulation
re_Ultuna	re_clim, 15 treatments x lenght of simulation years
J_Ultuna	Number of treatments

Table 2: The input data structure (here for Lanna)

Sheet name	Description
SOC_Lanna	SOC, 15 treatments x lenght of simulation years
SOC_init_Lanna	SOC initialization values, 15 treatments
error_SOC_Lanna	SOC mean errors, 15 treatments
error_SOC_multiplier_Lanna	SOC error weight
Yields_cereals_Lanna	Cereals yields, 8 treatments x lenght of simulation years
Straw_cereals_Lanna	Cereals straw, 8 treatments x lenght of simulation years
I_FYM_Lanna	Inputs farmyard manure, 15 treatments x lenght of simulation years
I_CO_Lanna	Inputs compost, 8 treatments x lenght of simulation years
I_SLU_Lanna	Inputs sludges, 8 treatments x lenght of simulation years
I_STR_Lanna	Inputs straw, 8 treatments x lenght of simulation years
N_Lanna	Lenght of the simulation
re_Lanna	re_clim, 8 treatments x lenght of simulation years
J_Lanna	number of treatments

Calibration (for Ultuna only)

I then run the calibration **only for Ultuna**

I first make some assumption about the initialization ratio priors, calculating them assuming average re values and average input values and calculating the steady states for a two pool model:

```
# INIT RATIO PRIORS
mean_re_Ultuna = mean(as.numeric(re_Ultuna_long[21, ]), na.rm = T)
mean_I_Ultuna = mean(Ultuna_yields_timeseries_long * 10^-3, na.rm = T)
YSS_Ultuna <- mean_I_Ultuna * 0.7/(0.8 * mean_re_Ultuna)
OSS_Ultuna <- 0.15 * (mean_I_Ultuna * 0.7/(0.0085 * mean_re_Ultuna))
Init_prior_Ultuna <- 1 - YSS_Ultuna/OSS_Ultuna
```

This value (0.9291667) is introduced in the calibration as average, with variance equal to $0.9291667 \cdot 0.2$

The MCMC model

The model, specified in JAGS, has a loop running for each treatment and a common section where the priors are defined. Here below I report only the prior section.

```
##Parameters Ultuna
# Xie, Yajun. 2020. "A Meta-Analysis of Critique of Litterbag Method Used in Examining Decomposition
k1_ult ~ dunif(0.78, 1)
k2_ult ~ dnorm(0.00605, 1/(0.00605*error_h)) T(0.00605-0.00605*0.5,0.00605+0.00605*0.5)

h_S_ult ~ dnorm(0.15,1/(0.15*error_h)) T(0.15-limits_h,0.15+limits_h)
h_R_ult ~ dnorm(0.35,1/(0.35*error_h)) T(0.35-limits_h, 1)
h_FYM_ult ~ dnorm(0.27,1/(0.27*error_h)) T(0.27-limits_h,0.27+limits_h)
h_PEA_ult ~ dnorm(0.59,1/(0.59*error_h)) T(0.59-limits_h, 0.59+limits_h)
h_SAW_ult ~ dnorm(0.25,1/(0.25*error_h)) T(0.25-limits_h,0.25+limits_h)
h_SLU_ult ~ dnorm(0.41,1/(0.41*error_h)) T(0.41-limits_h,0.41+limits_h)

#root/shoot ratios priors
SR_cereals_ult ~ dnorm(11, 1/(11*error_SR)) T(11-11*error_SR,11+11*error_SR)
SR_root_crops_ult ~ dnorm(29.49853, 1/(29.49853*error_SR)) T(29.49853-29.49853*error_SR,29.49853+29.49853*error_SR)
SR_oilseeds_ult ~ dnorm(8, 1/(8*error_SR)) T(8-8*error_SR,8+8*error_SR)
SR_maize_ult ~ dnorm(6.25, 1/(6.25*error_SR)) T(6.25-6.25*error_SR,6.25+6.25*error_SR)

exudates_coeff ~ dnorm(1.65,1/(1.65*0.1)) T(1.65*0.95,1.65*1.05)

Init_ratio_Ultuna ~ dnorm(0.9291667,1/(0.9291667*0.2)) T(0.8,0.98)

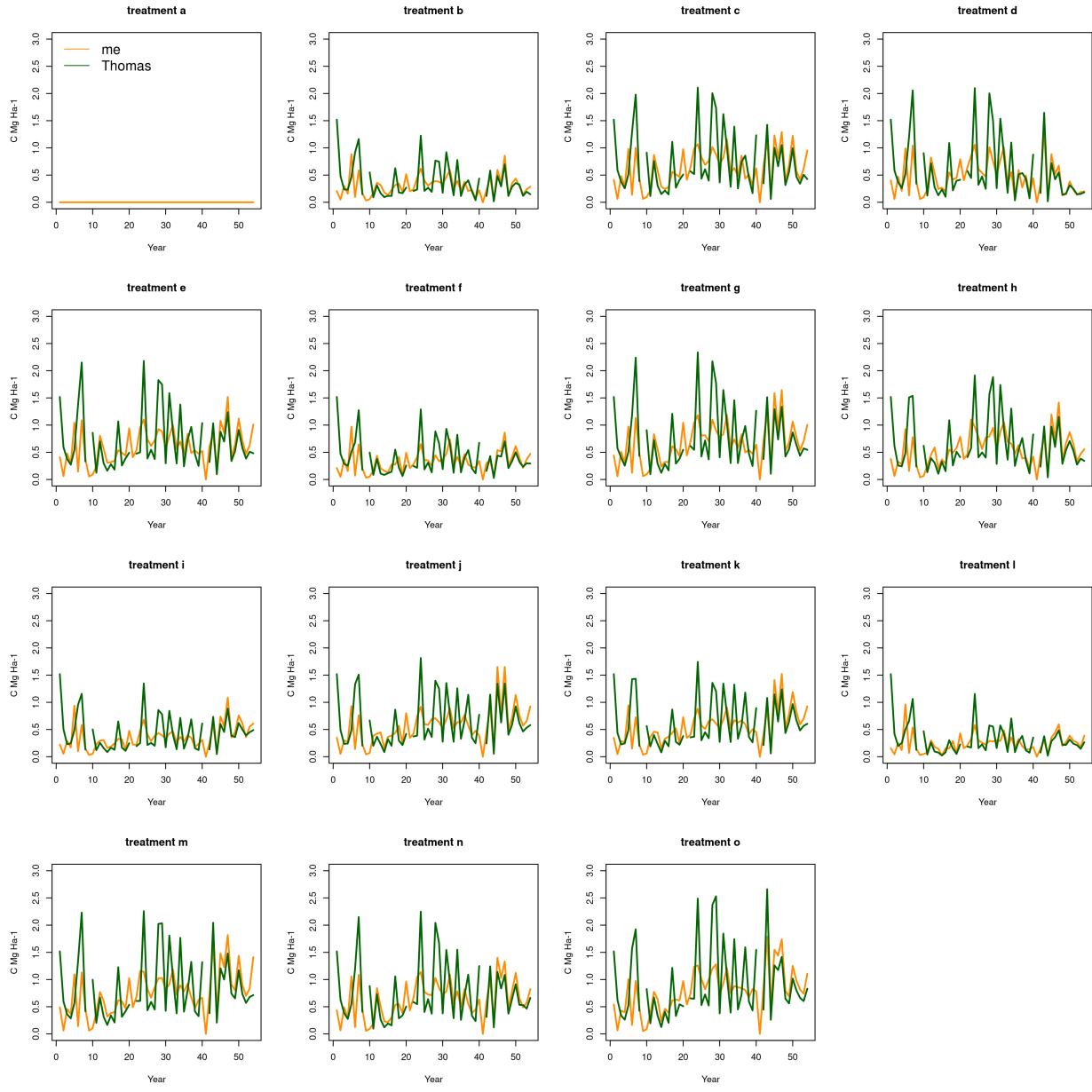
stubbles_ratio_Ultuna ~ dnorm(0.04,1/0.01) T(0.02,0.06)
stubbles_ratio_Lanna ~ dnorm(0.33,1/0.05) T(0.1,0.45)

C_percent ~ dunif(0.40, 0.51)

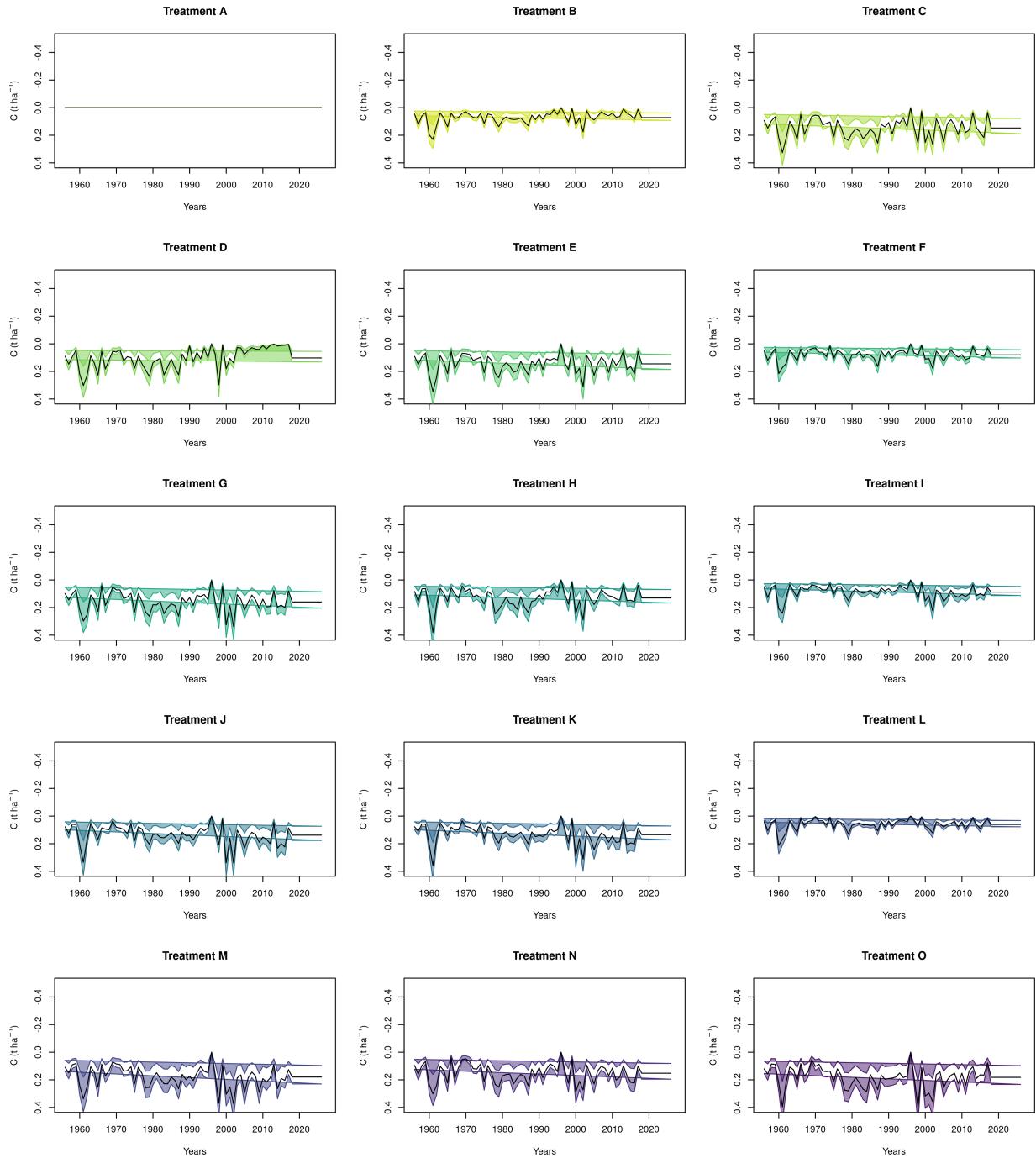
error_h<-0.2
limits_h<-0.3
error_SR<-0.25
```

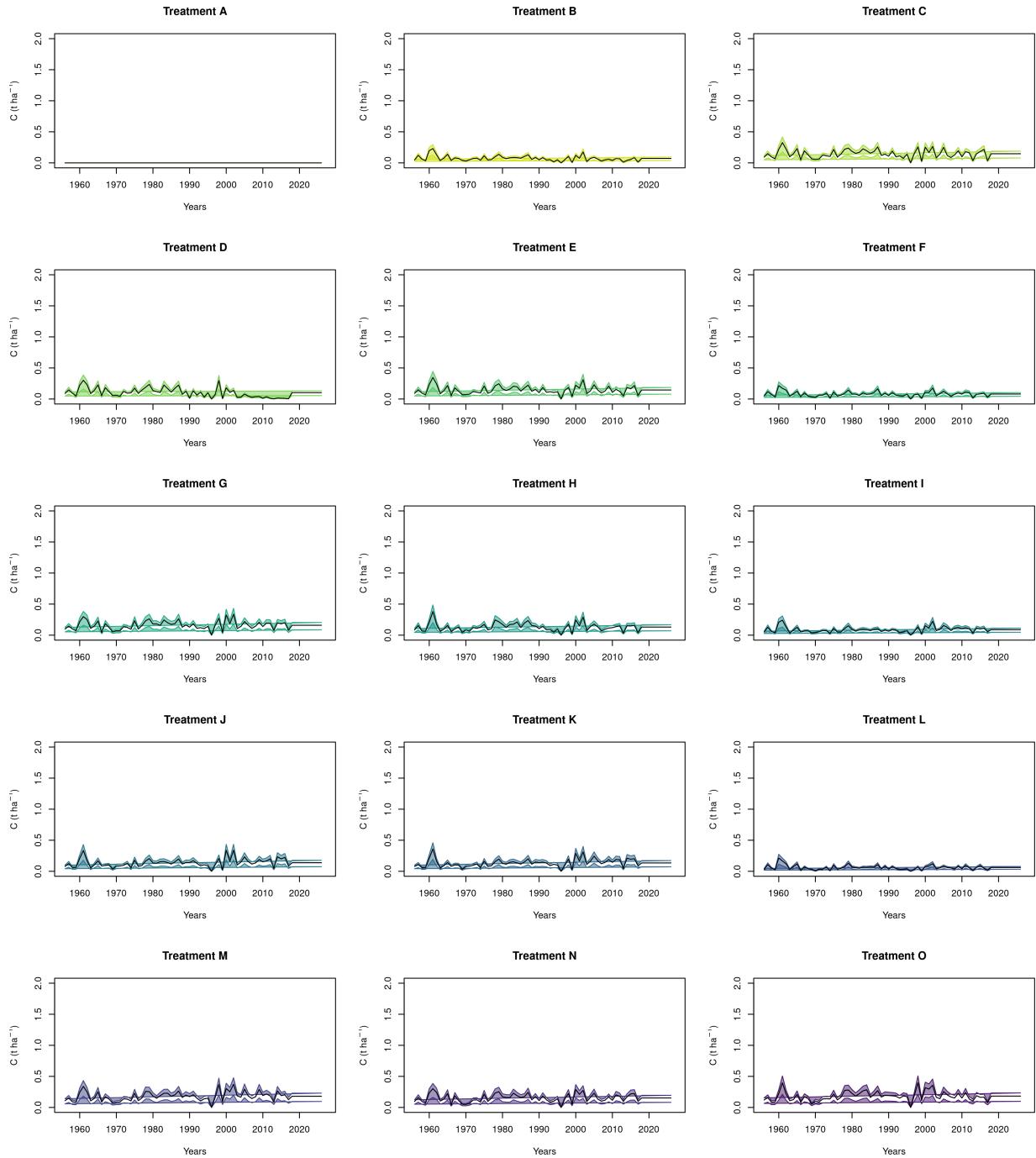
Inputs and other checks

The input calculation is within the Bayesian calibration, but I extract the mean value from the chains and compare it with former calculation (Thomas in 2011):



In the same way I can extract and plot separate components of the simulation (the averages shown here), like the fluxes, show here separated for roots and shoots:

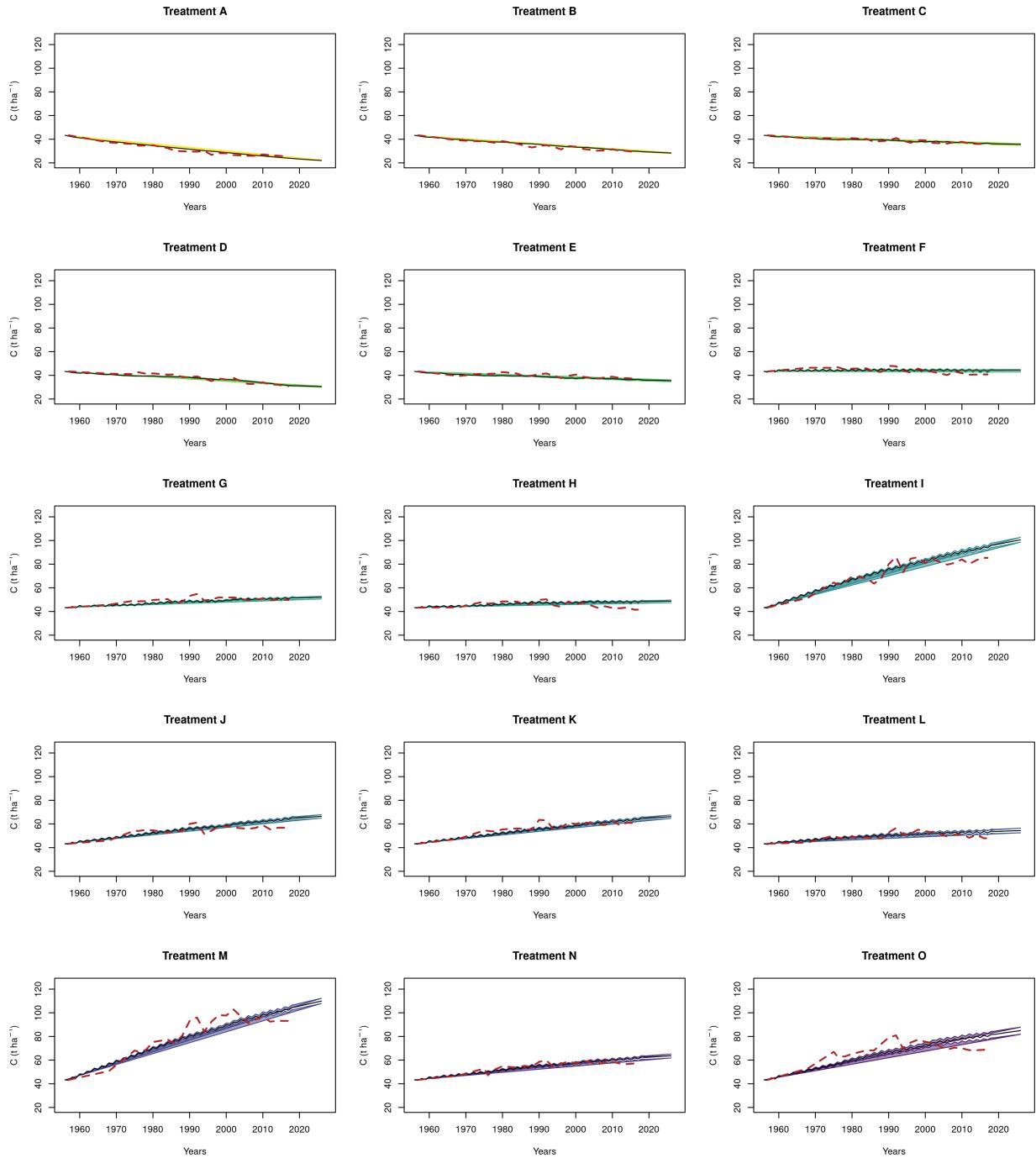




Remember here I call “fluxes” the term that sometimes is described as ϕ in Thomas literature, which collects many terms together: $\phi = h \cdot \frac{k_Y r_e Y_0 - i}{(k_O - k_Y) r_e}$
 (this written separately for all classes of inputs, for better description please refer to the Equation description file)

Predictions

Predictions do not seem to present any problem, but it is indeed pretty easy to calibrate a model with so much freedom, it always work:



Posteriors

The issue we need to work on is the posteriors, to have it compatible with the literature.

