

r_clim: calculating climate scaling factors for the ICBM SOC model

Lorenzo Menichetti

2023-09-06

1 Introduction

The package `r_clim` is a collection of functions to calculate the climate scaling effect. (due to soil moisture and temperature) on the decomposition of soil organic matter (SOC). The functions are intended as companion of the ICBM model (Kätterer and Andrén 2001) where they are used to rescale the two pools kinetics, but can be utilized to rescale any SOC decomposition process kinetic (for example as in Menichetti et al. (2019) where the same scaling has been applied to the microbial kinetic term in the Q decomposition model) or to normalize for the effects of such covariates in laboratory studies.

In this vignette we will first describe the steps of the calculation, and then introduce an example of possible usage, based on an example dataset.

2 The steps in the calculation

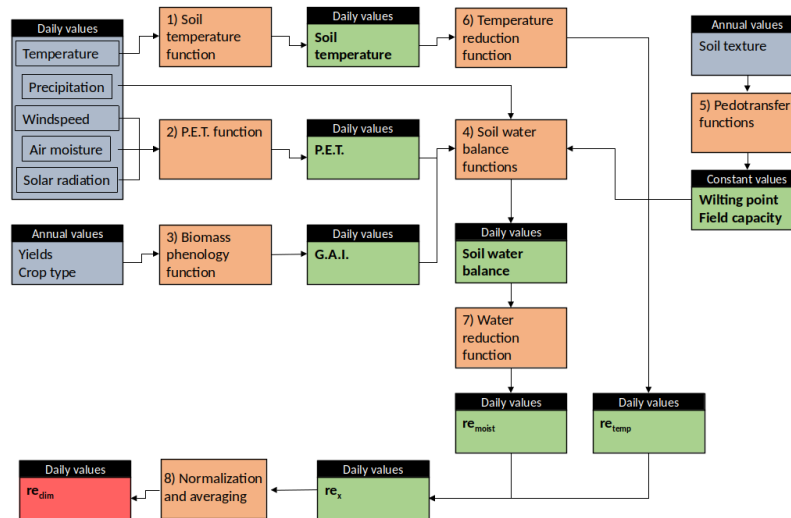


Figure 1: A bird's eye view of the procedure to calculate the climatic reduction coefficients

2.1 Soil temperature

2.2 Potential evapotranspiration

2.3 Biomass phenology functions (G.A.I. estimates)

2.4 Soil water balance

The steps of the calculation are already described in the package manual, but here we will examine some of them in more detail.

Step 1: Soil water W is initialized assuming saturation, based on the depth L and volumetric capacity
 $W[1] = \Theta_f \cdot L$

Step 2: The single crop coefficient K_c is calculated based on GAI

$$K_c = 1.3 - 0.5 \cdot \exp(-0.17 \cdot GAI)$$

Step 3: calculation of crop evapotranspiration (ET_c) under standard condition

$$ET_c = ET_0 \cdot K_c$$

Step 4: the intercepted water It is calculated based on crop ET , GAI and precipitation P

$$It = \min(P, ET_c, 0.2 \cdot GAI)$$

Step 5: potential evapotranspiration is calculated

$$E_{pot} = (ET_c - It)$$

Step 6: Calculation of the percolation. Water (W_b , water bypass) is lost when above field capacity, but allowing saturation for one day

$$W_b = \max(0, W - (\Theta_f \cdot L))$$

Step 7: Soil evaporation reduction coefficient

$$Kr = \left(1 - \frac{0.95 \cdot \Theta_{field} - \Theta}{0.95 \cdot \Theta_{field} - \alpha \cdot \Theta_{wilt}}\right)^2$$

This function expresses the actual evaporation, and in it it is considered that even below wilting point there is some biological activity. The function is zero below wilting point.

A subsequent conditions is applied so that Kr cannot be above one.

Step 8: Actual evapotranspiration is calculated

$$E_{act} = E_{pot} \cdot Kr$$

Step 9: The water balance is calculated (stepwise)

$$W[i + 1] = W[i] + P[i] - E_{act}[i] - It - W_b[i]$$

2.5 Pedotransfer functions

2.6 Temperature reduction function

2.7 Water reduction function

2.8 Normalization and averaging

2.9 Example

First we need to install the package from its GitHub repository, for which we need to rely on the package `devtools`.

We can then proceed to load the test site data and reorganize them averaging by treatment (since in this example we are interested in calculating the average r_{clim} factor. The test data are included in the `/data` folder of this package.

Now we run the `reclim` command to calculate the `r_clim` factor:

```
# #calculate the re_clim (daily values, all treatments)
```

```

Test_reclim_out<-reclim(weather=weather_testdata,
                        aboveground=aboveground_testdata,
                        latitude=58,
                        altitude=83,
                        sand=8,
                        clay=43,
                        ave_SOC=1.9,
                        depth=20,
                        sun.mode="Rsolar")
#> Hi there! we're going to try our best to calculate your ICBM climatic reduction factors...
#> please forgive me for any issue, and write me an email at ilmenichetti@gmail.com, we can maybe try
#>
#> performing basic data integrity check on weather data
#> ## basic data integrity check on weather data cleared
#> performing basic data integrity check on aboveground data
#> ## basic data integrity check on aboveground data cleared
#> performing data integrity check on the consistency of the two dataset
#> ## datasets appear to cover the same period
#> checking if basic soil phisical data are available
#> ## basic soil phisical data are present
#> porosity calculation
#> wilting point calculation
#> field capacity calculation
#> PET calculation
#> sun mode: Rsolar - calculating from directly measured Rsolar
#>
#> performing calculations for treatment CONVENTIONAL
#> GAI calculation for treatment CONVENTIONAL
#> Soil T calculation for treatment CONVENTIONAL
#> Water balance calculation for treatment CONVENTIONAL
#> re_wat calculation for treatment CONVENTIONAL
#> re_temp calculation for treatment CONVENTIONAL
#>
#> performing calculations for treatment ORGANIC
#> GAI calculation for treatment ORGANIC
#> Soil T calculation for treatment ORGANIC
#> Water balance calculation for treatment ORGANIC
#> re_wat calculation for treatment ORGANIC
#> re_temp calculation for treatment ORGANIC

```

Ok, the calculation seems to be running.

The output of the function is fairly complicated list with most outputs from the internal functions. We can for eample have a look at the structure of the rescaled r_{clim} value, which is usually the main target value:

```

str(Test_reclim_out$re_x1)
#> num [1:2, 1:5114] 0.0125 0.0125 0.0476 0.0476 0.0381 ...
#> - attr(*, "dimnames")=List of 2
#> ..$ : chr [1:2] "CONVENTIONAL" "ORGANIC"
#> ..$ : NULL

```

Now we can try to plot some of the values we calculated, for example the GAI simulation:

```

crop_id_used<-unique(Test_reclim_out$crop_id)
palette_crop_id_used<-brewer.pal(length(crop_id_used), "Dark2")

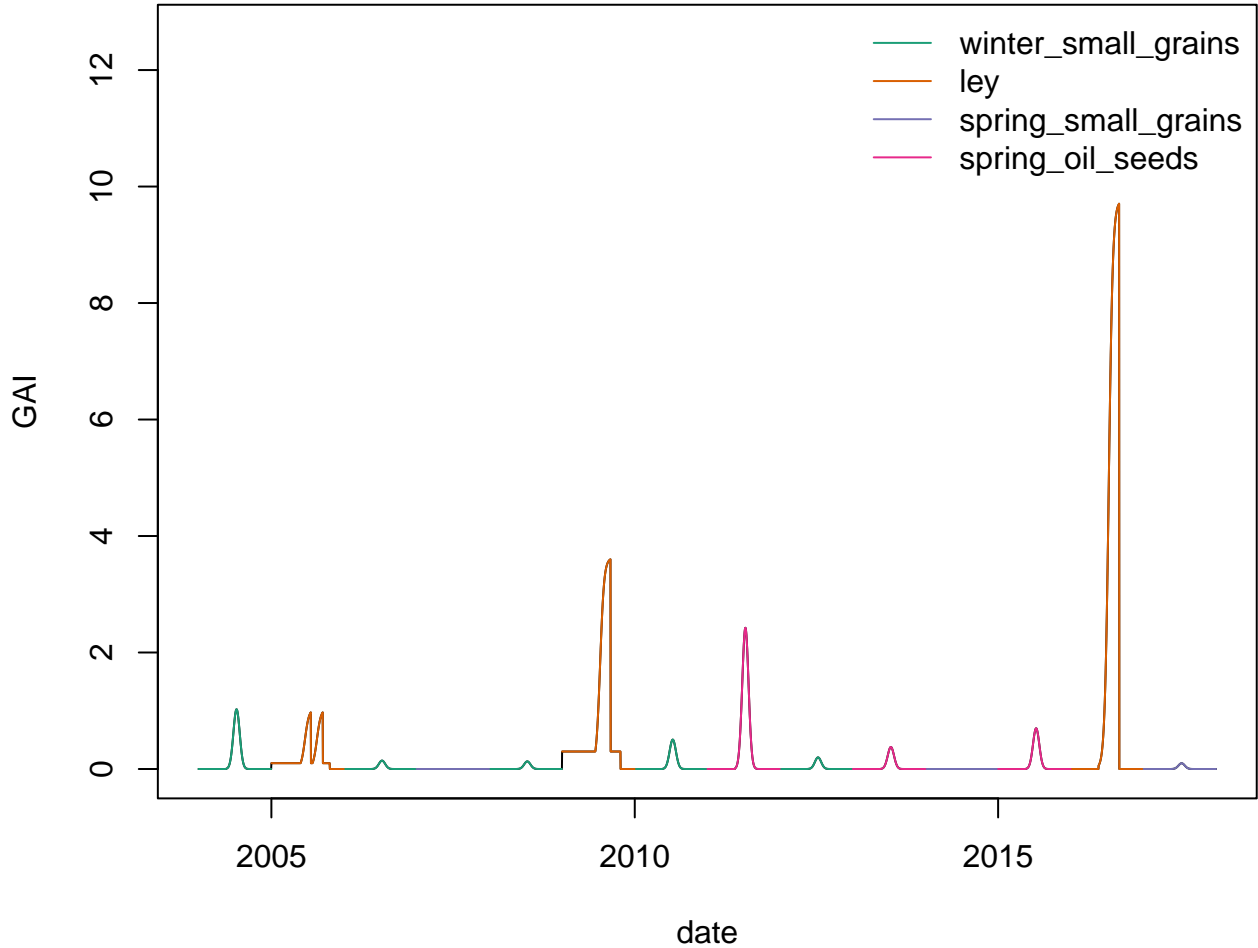
```

```

plot(Test_reclim_out$PET$date, Test_reclim_out$GAI[2,], type="l", ylim=c(0, max(Test_reclim_out$GAI[2,]

#coloring the output based on the crop
for(i in 1:length(crop_id_used)){
  which_ones<-which(!Test_reclim_out$crop_id==crop_id_used[i])
  GAI_crop<-Test_reclim_out$GAI[2,]
  GAI_crop[which_ones]<-NA
  lines(Test_reclim_out$PET$date, GAI_crop, col=palette_crop_id_used[i])
}
legend("topright", as.character(crop_id_used), col=palette_crop_id_used, bty="n", lty=1)

```

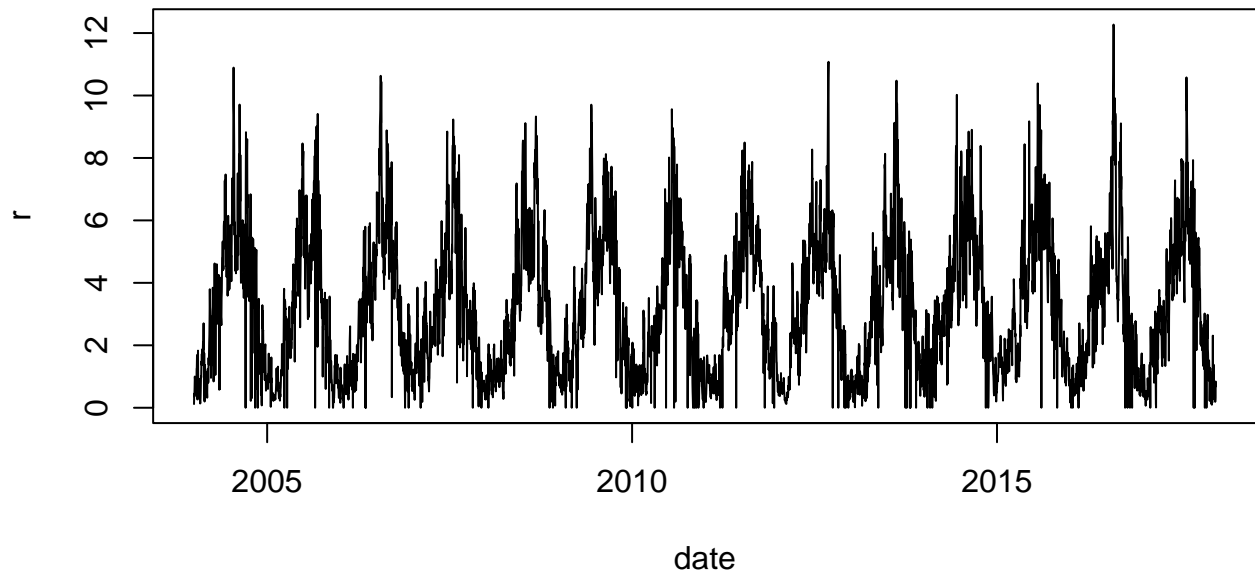


```

plot(Test_reclim_out$PET$date, Test_reclim_out$re_crop[1,], type="l",xlab="date", ylab="r", main="First

```

First treatment (CONVENTIONAL)

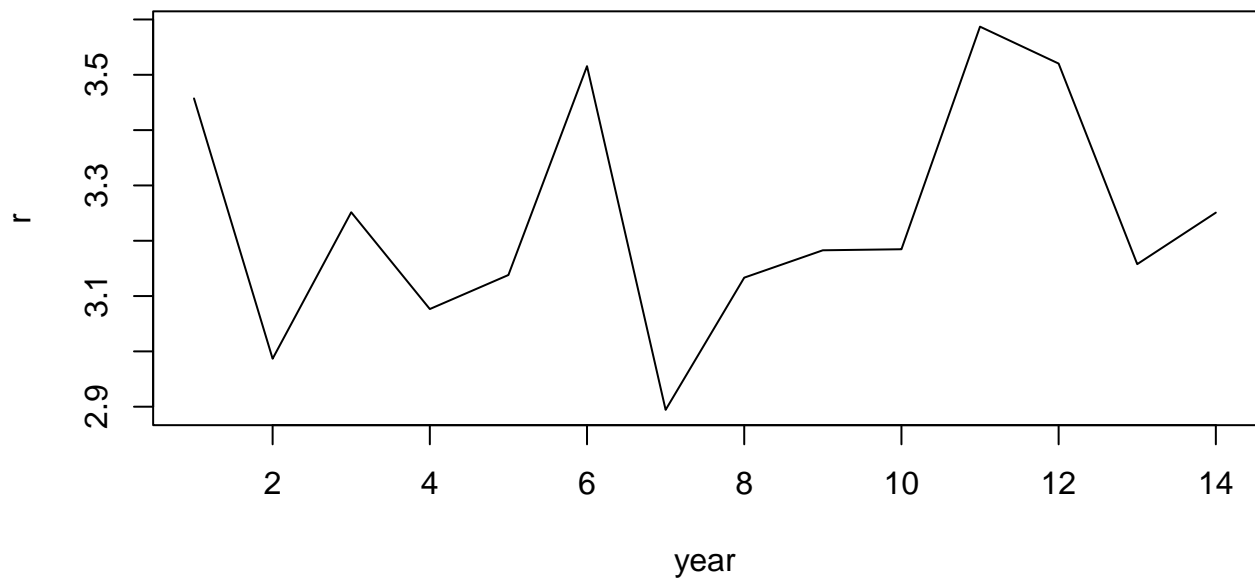


The package includes also a function to calculate the annual averages from the inputs of the main function:

```
#calculate annual re-clim values
Test_annual<-reclim_annual(Test_reclim_out$results_daily)
where_re_crop<-grepl("re_crop.", as.character(colnames(Test_annual)))
Test_annual_re_crop<-(Test_annual[where_re_crop])

plot(Test_annual_re_crop$re_crop_treat.CONVENTIONAL, type="l",xlab="year", ylab="r", main="First treatment (CONVENTIONAL)")
```

First treatment (CONVENTIONAL)



References

- Kätterer, Thomas, and Olof Andrén. 2001. “The ICBM Family of Analytically Solved Models of Soil Carbon, Nitrogen and Microbial Biomass Dynamics — Descriptions and Application Examples.” *Ecological Modelling* 136 (2-3): 191–207. [https://doi.org/10.1016/s0304-3800\(00\)00420-8](https://doi.org/10.1016/s0304-3800(00)00420-8).
- Menichetti, Lorenzo, Göran I. Ågren, Pierre Barré, Fernando Moyano, and Thomas Kätterer. 2019. “Generic Parameters of First-Order Kinetics Accurately Describe Soil Organic Matter Decay in Bare Fallow Soils over a Wide Edaphic and Climatic Range.” *Scientific Reports* 9 (1): 20319. <https://doi.org/10.1038/s41598-019-55058-1>.