Supplement S1 - Simulations material and method

Rémy Beugnon, Nolwenn Le Guyader, Alexandru Milcu, Jonathan Lenoir Jérémy Puissant, Xavier Mori

2023-11-08

We performed modelling to illustrate the consequences of vegetation diversity effects on microclimate and their consequences for ecosystems functioning. In this example climate was simplified to temperature.

First, we retrieved air temperature from a locality close to Jena (Germany) using ERA5 hourly data Copernicus Climate Change Service (n.d.). Second, using the model first from Huang et al. (2023), we predicted soil temperature for grassland field (with a diversity gradient: no plant, monoculture, polyculture of 60 plant species, (Huang et al. 2023)). Third, using the soil respiration - soil temperature relationship measured in grassland (Jones et al. 2006), we predicted soil respiration over time, for each microclimate driven by a specific level of diversity.

```
## R version 4.3.1 (2023-06-16)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.3.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.0   dplyr_1.1.3
##  [5] purrr_1.0.2     tidyr_1.3.0     tibble_3.2.1    tidyverse_2.0.0
##  [9] readr_2.1.4     zoo_1.8-12      mgcv_1.9-0      nlme_3.1-163
## [13] ggpubr_0.6.0    ggplot2_3.4.4
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    rstatix_0.7.2     stringi_1.7.12
##  [5] lattice_0.21-9    hms_1.1.3         digest_0.6.33     magrittr_2.0.3
##  [9] timechange_0.2.0  evaluate_0.22     grid_4.3.1        fastmap_1.1.1
## [13] Matrix_1.6-1.1    backports_1.4.1   fansi_1.0.5       scales_1.2.1
## [17] abind_1.4-5       cli_3.6.1         rlang_1.1.1       munsell_0.5.0
## [21] splines_4.3.1     withr_2.5.1       yaml_2.3.7        tools_4.3.1
## [25] tzdb_0.4.0        ggsignif_0.6.4    colorspace_2.1-0  broom_1.0.5
## [29] vctrs_0.6.4       R6_2.5.1          lifecycle_1.0.3   car_3.1-2
## [33] pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.4      glue_1.6.2
```

```
## [37] xfun_0.40         tidyselect_1.2.0  rstudioapi_0.15.0 knitr_1.44
## [41] htmltools_0.5.6.1 rmarkdown_2.25    carData_3.0-5     compiler_4.3.1
```

Data

Temperature data

First, we retrieved the air temperature of Jena from the ERA5 hourly dataset Copernicus Climate Change Service (n.d.).

```
jena_data<-read_table("Jena_JEclimateStation.csv") # retrieving data
air_temp_1=jena_data|> # renaming columns
  mutate("utc_date"=`""utc_date""`)|>
  mutate("utc_time"=`""utc_time""`)|>
  mutate("airT"= `""T_air""`)|>
  select(utc_date, utc_time, airT) #keeping only the columns of interest
air_temp=air_temp_1|>
  mutate("datetime"=
           as.POSIXct(paste(air_temp_1$utc_date, air_temp_1$utc_time),
                                format="%Y-%m-%d %H:%M:%S", tz="UTC"))|>
  #merging date and time columns
  select(datetime, airT)

air_temp_data= air_temp|> # splitting the date information
  # add months
  mutate( month= format(as.Date(air_temp$datetime,
                                 format="%d/%m/%Y"),"%m")) |>
  # add hours
  mutate( hour= format(as.POSIXct(air_temp$datetime,
                                   format="%d/%m/%Y %H"),"%H")) |>
  #add seasons
  mutate(season=if_else(month %in% c("09", "10", "11"), "autumn",
                        if_else(month %in% c("03", "04", "05"), "spring",
                                if_else(month %in% c("06", "07", "08"),"summer",
                                        "winter")))
  )|>
  mutate(year=year(datetime))|>
  select(datetime, year, season,month, hour,  airT)
```

Air temperature data over time in Jena:

```
## # A tibble: 6 x 6
##   datetime             year season month hour   airT
##   <dttm>              <dbl> <chr>  <chr> <chr> <dbl>
## 1 2006-01-01 00:00:00  2006 winter 01    00     2.60
## 2 2006-01-01 01:00:00  2006 winter 01    01     2.2
## 3 2006-01-01 02:00:00  2006 winter 01    02     1.44
## 4 2006-01-01 03:00:00  2006 winter 01    03     0.15
## 5 2006-01-01 04:00:00  2006 winter 01    04     0.55
## 6 2006-01-01 05:00:00  2006 winter 01    05     1.72
```

Then, we used the Jena Experiment data (Huang et al. 2023) to determine the offset between air and soil temperature.

```
#get all the files in the project
files=list.files()
```

```r
#select the digitalized for the offset between air and soil temperatures
a_s_T=files[grepl("offset_air_soil", files)]
seasons=str_sub(a_s_T, 17L, -5L)
#Get the number of points per season
L=a_s_T|>
  map(read.csv)
l=L|>
  map(length(L[[1]]))
l=l|>
  map(length) #number of points for each season
#create a data frame with all the seasons
df_off_as=a_s_T|>
  map_df(read.csv)
df_off_as=df_off_as|>
  #rename the two variables extracted on graph
  rename("hour_of_day"="x","temp_offset"="y")|>
  #create a factor column season
  mutate(season=as.factor(rep(seasons, l)))

#fitting a model for each level of diversity in function of the season
fit_hour_offset_as =
  mgcv::gam(temp_offset ~ season + s(hour_of_day, by = season),
            data=df_off_as)

air_soil_temp_offset<-function(hour, season){
  pred_df = data.frame(hour_of_day = hour, season = season) #create a data frame
  #in the same format as the one used for the predictions
  mod=fit_hour_offset_as #selection of the model according to the diversity level
  pred=predict(mod, newdata=pred_df) # prediction of the temperature offset
  return (pred)
}
```

Afterwards, we calculated the soil temperature corresponding to the air temperature.

```r
f_soilT<-function(airT, Dhour, season){
  #air temperature, hour of the day and season given
  offset_temp = as.vector(air_soil_temp_offset(as.numeric(Dhour), season))
  soilT=airT+offset_temp #calculating the soil temperature
  return(soilT)
}

soilT=f_soilT(air_temp_data$airT, air_temp_data$hour, air_temp_data$season)
#for each air temperature we got the soil temperature
#adding soil temperature in the data frame
air_soil_temp_data=cbind(air_temp_data, soilT)
```

Air and soil temperatures data over time in Jena:

```
##              datetime year season month hour  airT    soilT
## 1 2006-01-01 00:00:00 2006 winter    01   00 2.605 3.058652
## 2 2006-01-01 01:00:00 2006 winter    01   01 2.200 2.617469
## 3 2006-01-01 02:00:00 2006 winter    01   02 1.435 1.844843
## 4 2006-01-01 03:00:00 2006 winter    01   03 0.150 0.602237
## 5 2006-01-01 04:00:00 2006 winter    01   04 0.550 1.091147
## 6 2006-01-01 05:00:00 2006 winter    01   05 1.715 2.360105
```

Diversity effect on microclimate

Based on (Huang et al. 2023), we determined the effect of diversity on the temperature.

```
#get all the files in the project
files=list.files()
#select the digitalized for the Monoculture
div1=files[grepl("div_1.csv", files)]
#select the digitalized for the Polyculture
div60=files[grepl("div_60.csv", files)]
#create a vector with the four seasons
seasons=str_sub(div1, 1L, -11L)
#For the Monoculture (diversity of one species)
#Get the number of points per season
L.1=div1|>
  purrr::map(read.csv)
l1=L.1|>
  purrr::map(length(L.1[[1]]))
l1=l1|>
  purrr::map(length) #number of points for each season
#create a data frame with all the seasons
df.1=div1|>
  purrr::map_df(read.csv)
df.1=df.1|>
  #rename the two variables extracted on graph
  rename("hour_of_day"="x","temp_offset"="y")|>
  #create a factor column season
  mutate(season=as.factor(rep(seasons, l1)))|>
  #create a factor column diversity
  mutate(diversity=as.factor(rep(1, nrow(df.1))))

#For the Polyculture (diversity f 60 species)
#Get the number of points per season
L.60=div60|>
  purrr::map(read.csv)
l60=L.60|>
  purrr::map(length(L.60[[1]]))
l60=l60|>
  purrr::map(length) #number of points for each season
#create a data frame with all the seasons
df.60=div60|>
  purrr::map_df(read.csv)
df.60=df.60|>
  #rename the two variables extracted on graph
  rename("hour_of_day"="x","temp_offset"="y")|>
  #create a factor column season
  mutate(season=as.factor(rep(seasons, l60)))|>
  #create a factor column diversity
  mutate(diversity=as.factor(rep(60, nrow(df.60))))

#data frame with all the data
df=rbind(df.1, df.60)
#fitting a model for each level of diversity in function of the season
fit_hour_offset=
  purrr::map(list(df.1, df.60),
```

```r
      ~list(model = mgcv::gam(temp_offset~season+s(hour_of_day, by=season),
                              data=.x),
            data = .x))
#Function that gives the microclimate offset
micro_temp_offset<-function(hour, season, diversity){
  div=which(levels(df$diversity)==diversity) #getting the level of diversity to
  #select the right model in the list
  pred_df=data.frame(hour_of_day=hour, season= season, diversity=diversity)
  #in the same format as the one used for the predictions
  mod=fit_hour_offset[[div]] #selection of the model according to the diversity level
  pred=predict(mod$model, newdata=pred_df) # prediction of the temperature offset
  return (pred)
}
#Function that gives the micro temperature for a given macrotemperature
f_microT<-function(macroT, Dhour, season, div){
  offset_temp = as.vector(micro_temp_offset(as.numeric(Dhour), season, div))
  microT=macroT+offset_temp
  return(microT)
}

#For the monoculture we have for temperature:
div1= 1
microT1=f_microT(air_soil_temp_data$soilT,
                 air_soil_temp_data$hour,
                 air_soil_temp_data$season,
                 div1)
#For the polyculture we have for temperature:
div60= 60
microT60=f_microT(air_soil_temp_data$soilT,
                  air_soil_temp_data$hour,
                  air_soil_temp_data$season,
                  div60)

#Data frame with the four different temperature at each time step
all_temp_data=air_soil_temp_data|>
  mutate(microT1=microT1)|>
  mutate(microT60=microT60)
```

Air temperature (airT), soil temperature (soilT) and microtemperatures from a monoculture (microT1) and a polyculture (microT60) in Jena grassland:

```
##              datetime year season month hour  airT    soilT   microT1 microT60
## 1 2006-01-01 00:00:00 2006 winter    01   00 2.605 3.058652 3.1555317 3.746892
## 2 2006-01-01 01:00:00 2006 winter    01   01 2.200 2.617469 2.7203346 3.314877
## 3 2006-01-01 02:00:00 2006 winter    01   02 1.435 1.844843 1.9501727 2.550054
## 4 2006-01-01 03:00:00 2006 winter    01   03 0.150 0.602237 0.7042429 1.313224
## 5 2006-01-01 04:00:00 2006 winter    01   04 0.550 1.091147 1.1851978 1.806693
## 6 2006-01-01 05:00:00 2006 winter    01   05 1.715 2.360105 2.4451463 3.080082
```

Temperature-dependent ecosystem function: soil respiration

From (Jones et al. 2006) we got the soil respiration in function of the soil temperature, which allows us to determine the soil respiration for each microclimate (bare soil, monoculture, polyculture).

```
response_to_temperature <- function (temp, funct){ #temperature and the ecosystem
                                                  #function chosen
  df=data_frame(temperature= temp)
  pred= predict(funct,df)
  vpred=as.vector(pred)
  return(vpred)
}

variable <- "soilresp" #variable that changes with temperature
file<-paste0(variable, "_temp_data.csv") #name of the data files containing
                                        #the value of the variable at given
                                        #temperatures
tab_resp_temp<-read.csv(file) #creation of the data frame
colnames(tab_resp_temp)<-c("temperature", variable) # changing the columns' names
function_resp_temp<-gam(soilresp~s(temperature), data=tab_resp_temp)
```

Soil respiration (in $g\ CO_2.m^{-2}.h^{-1}$) in function of soil temperature (°C):

```
##   temperature  soilresp
## 1    1.375000 0.2102273
## 2    1.589286 0.2215909
## 3    1.839286 0.2159091
## 4    2.053571 0.2272727
## 5    2.250000 0.2272727
## 6    2.446429 0.2386364
```

Prediction of ecosystem functioning for the different levels of diversity

For each microclimate (bare soil, monoculture and polyculture), we predicted the soil respiration.

```
# Getting the response to temperature corresponding to each temperature
Lresp=purrr::map(list(all_temp_data$soilT,
               all_temp_data$microT1,
               all_temp_data$microT60),
          ~response_to_temperature(.x, function_resp_temp))

# adding them in a dataframe
dfresp=as.data.frame(do.call(cbind, Lresp))
#arrange the dataframe
dfresp=dfresp|>
  rename("soilrespsoilT"="V1", "soilrespmicroT1"="V2", "soilrespmicroT60"="V3")
```

We created a data frame containing all the data and for a few years only.

```
#Add the respiration in the temperature dataframe
df_resp_to_microclimate= bind_cols(all_temp_data, dfresp)
df_resp_to_microclimate=df_resp_to_microclimate|>
  arrange(datetime)|>
  filter(!is.na(datetime))

#Building a final table over three years starting in the growing season
df_resp_to_microclimate_growing= df_resp_to_microclimate |>
  # april is assimilated to the beginning of spring, i.e. the growing season
```

```r
  filter(datetime>="2017-04-01")|>
  # calculating the cumulated respiration under
  # the soil temperature over the whole time period
  mutate(cumsoilrespsoilT=cumsum(soilrespsoilT))|>
  # for the microtemperture of the monoculture
  mutate(cumsoilrespmicroT1=cumsum(soilrespmicroT1))|>
  # for the microtemperture of the polyculture
  mutate(cumsoilrespmicroT60=cumsum(soilrespmicroT60))
```

Final table gathering information about the date (datetime, year, season, month, hour), the temperatures (airT, soilT, microT1, microT60), the corresponding soil respirations (soilrespsoilT, soilrespmicroT1, soilrespmicroT60), and the corresponding cumulated soil respirations over the whole period considered (cumsoilrespsoilT, cumsoilrespmicroT1, cumsoilrespmicroT60):

```
##              datetime year season month hour  airT    soilT  microT1 microT60
## 1 2017-03-31 22:00:00 2017 spring    03   22 6.870 9.466981 9.296067 9.074125
## 2 2017-03-31 23:00:00 2017 spring    03   23 6.165 8.535407 8.447783 8.421922
## 3 2017-04-01 00:00:00 2017 spring    04   00 5.705 8.570513 8.622824 8.870340
## 4 2017-04-01 01:00:00 2017 spring    04   01 5.235 8.149670 8.210692 8.551481
## 5 2017-04-01 02:00:00 2017 spring    04   02 6.220 9.209395 9.285374 9.719773
## 6 2017-04-01 03:00:00 2017 spring    04   03 4.690 7.752648 7.852399 8.374722
##   soilrespsoilT soilrespmicroT1 soilrespmicroT60 cumsoilrespsoilT
## 1     0.5179863       0.5083496        0.4960957        0.5179863
## 2     0.4674255       0.4628906        0.4615584        0.9854118
## 3     0.4692518       0.4719835        0.4850819        1.4546636
## 4     0.4477021       0.4507819        0.4682610        1.9023657
## 5     0.5035306       0.5077527        0.5325898        2.4058963
## 6     0.4280070       0.4329009        0.4591345        2.8339033
##   cumsoilrespmicroT1 cumsoilrespmicroT60
## 1          0.5083496           0.4960957
## 2          0.9712402           0.9576542
## 3          1.4432236           1.4427361
## 4          1.8940056           1.9109971
## 5          2.4017582           2.4435869
## 6          2.8346592           2.9027214
```

Then, we created a specific table to focus on the soil respiration over a month scale for the first year of data considered (2017-2018).

```r
# Respiration table for months scale
df_resp_month=df_resp_to_microclimate_growing|>
  filter(year=="2017"|
           grepl("2018-03", datetime)|
           grepl("2018-02", datetime)|
           grepl("2018-01", datetime))|> #keeping the first year of data
  group_by(month(datetime))|> #grouping by months
  rename("mt"="month(datetime)")|>
    #taking for each respiration the monthly mean and the 5% and 95% quantiles
  summarise(mmaT=mean(soilrespsoilT),
            mmiT1=mean(soilrespmicroT1),
            mmiT60=mean(soilrespmicroT60),
            fqmaT=quantile(soilrespsoilT, probs = 0.05),
            fqmiT1=quantile(soilrespmicroT1, probs =0.05),
            fqmiT60=quantile(soilrespmicroT60, probs = 0.05),
            lqmaT=quantile(soilrespsoilT, probs =  0.95),
```

```
            lqmiT1=quantile(soilrespmicroT1, probs =0.95),
            lqmiT60=quantile(soilrespmicroT60, probs = 0.95))
```

```
## # A tibble: 6 x 10
##      mt  mmaT mmiT1 mmiT60   fqmaT  fqmiT1  fqmiT60 lqmaT lqmiT1 lqmiT60
##   <dbl> <dbl> <dbl>  <dbl>   <dbl>   <dbl>    <dbl> <dbl>  <dbl>   <dbl>
## 1     1 0.316 0.318  0.335  0.174   0.175    0.186  0.553  0.557   0.591
## 2     2 0.120 0.121  0.133 -0.0478 -0.0456 -0.0303 0.265  0.266   0.278
## 3     3 0.335 0.318  0.288  0.0334  0.0296  0.0174 0.689  0.655   0.610
## 4     4 0.562 0.537  0.493  0.252   0.246    0.229  1.04   0.959   0.812
## 5     5 1.21  1.16   1.08   0.450   0.438    0.373  2.47   2.36    2.18
## 6     6 1.82  1.67   1.54   1.01    0.937    0.873  2.89   2.62    2.36
```

Similarly, we created a specific table to focus on the soil respiration over over a years scale.

```
Lyear=c("2017","2018","2019")

# Respiration table for years scale
L_resp_year=map(.x=Lyear, #for each year
       .f=~
         (df_resp_to_microclimate_growing|>
  filter(year==.x)|>
  group_by(month(datetime))|> #grouping by months
    # getting the mean monthly respiration and the quantiles at 5% and 95%
  rename("mt"="month(datetime)")|>
  summarise(
    mmaT=mean(soilrespsoilT),
    mmiT1=mean(soilrespmicroT1),
    mmiT60=mean(soilrespmicroT60),
    fqmaT=median(soilrespsoilT[soilrespsoilT>=
                                 quantile(soilrespsoilT, probs=0.95)]),
    fqmiT1=median(soilrespmicroT1[soilrespmicroT1>=
                                    quantile(soilrespmicroT1, probs=0.95)]),
    fqmiT60=median(soilrespmicroT60[soilrespmicroT60>=
                                      quantile(soilrespmicroT60, probs=0.95)]),
    lqmaT=median(soilrespsoilT[soilrespsoilT<=
                                 quantile(soilrespsoilT, probs=0.05)]),
    lqmiT1=median(soilrespmicroT1[soilrespmicroT1<=
                                    quantile(soilrespmicroT1, probs=0.05)]),
    lqmiT60=median(soilrespmicroT60[soilrespmicroT60<=
                                      quantile(soilrespmicroT60, probs=0.05)])
    )|>
  mutate(year=.x)) #adding a year variable
       )
df_resp_year=as.data.frame(do.call(rbind, L_resp_year))

df_resp_year=df_resp_year|>
  unite(datetime, c("year", "mt"), sep="-") # uniting by date

#removing the first line that is wrongly assess as the 31st of March
df_resp_year=df_resp_year[-1,]

df_resp_year=df_resp_year|>
  mutate(datetime=as.Date(as.yearmon(datetime))) #creating a month year variable
```

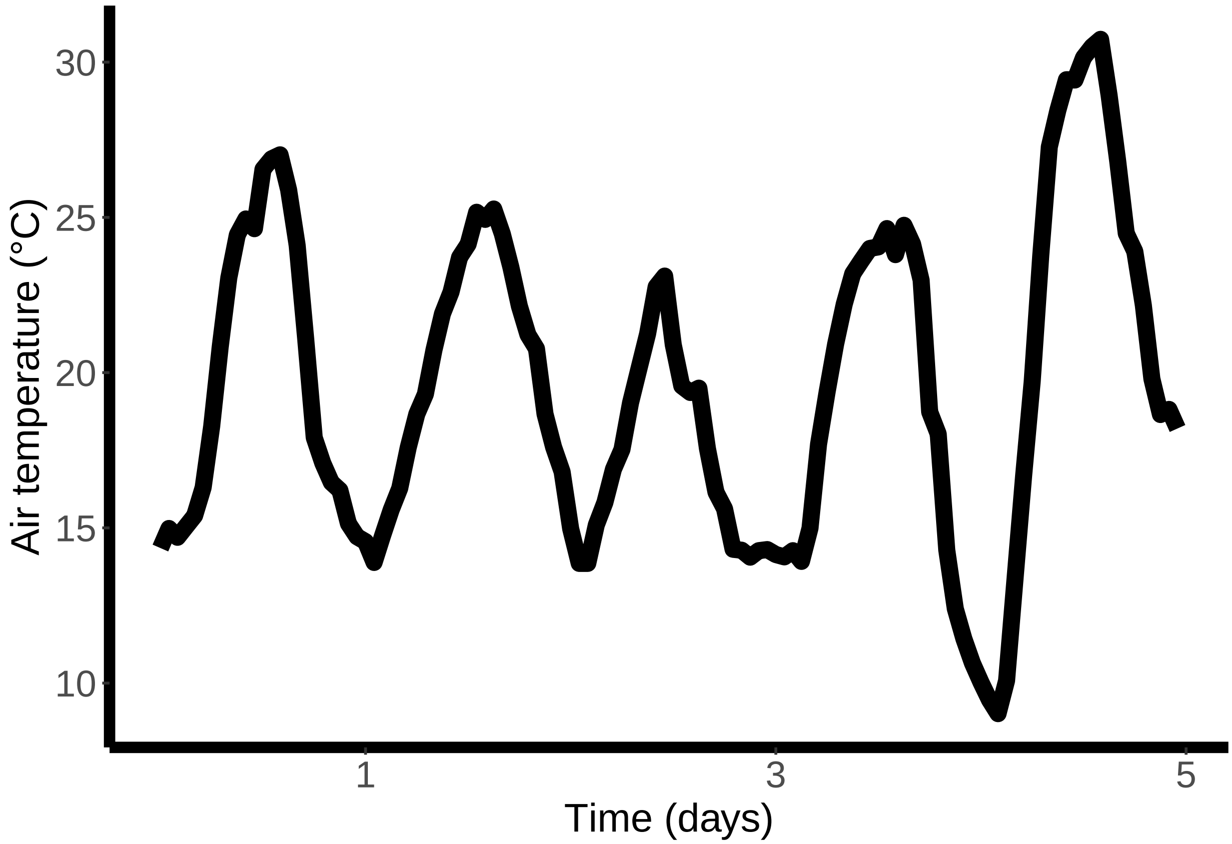
```
##     datetime      mmaT     mmiT1   mmiT60    fqmaT   fqmiT1    fqmiT60     lqmaT
## 2 2017-04-01 0.5616915 0.5368210 0.492630 1.296041 1.208044 0.9866678 0.2268052
## 3 2017-05-01 1.2065082 1.1640754 1.081547 2.671724 2.603748 2.4620779 0.3409402
## 4 2017-06-01 1.8171366 1.6717386 1.543944 3.123566 2.891645 2.5618243 0.9041966
## 5 2017-07-01 1.9395214 1.7927961 1.665037 3.132160 2.863174 2.6740267 1.0336313
## 6 2017-08-01 1.8578820 1.7136349 1.586740 3.143918 2.874561 2.6008483 0.8085434
## 7 2017-09-01 0.8961075 0.9042066 0.943449 1.707100 1.689971 1.6574925 0.3984556
##      lqmiT1   lqmiT60
## 2 0.2242760 0.2069705
## 3 0.3460862 0.3281463
## 4 0.8456262 0.8036819
## 5 0.9895375 0.8477522
## 6 0.7299965 0.6843879
## 7 0.4112238 0.4450658
```

Plots

```r
my_theme<- theme_classic()+theme(axis.line.x = element_line(size=2),
                                 axis.line.y=element_line(size=2),
        axis.text.x = element_text(size=14), axis.text.y=element_text(size=14),
        axis.title.x = element_text(size=15), axis.title.y=element_text(size=15))
```

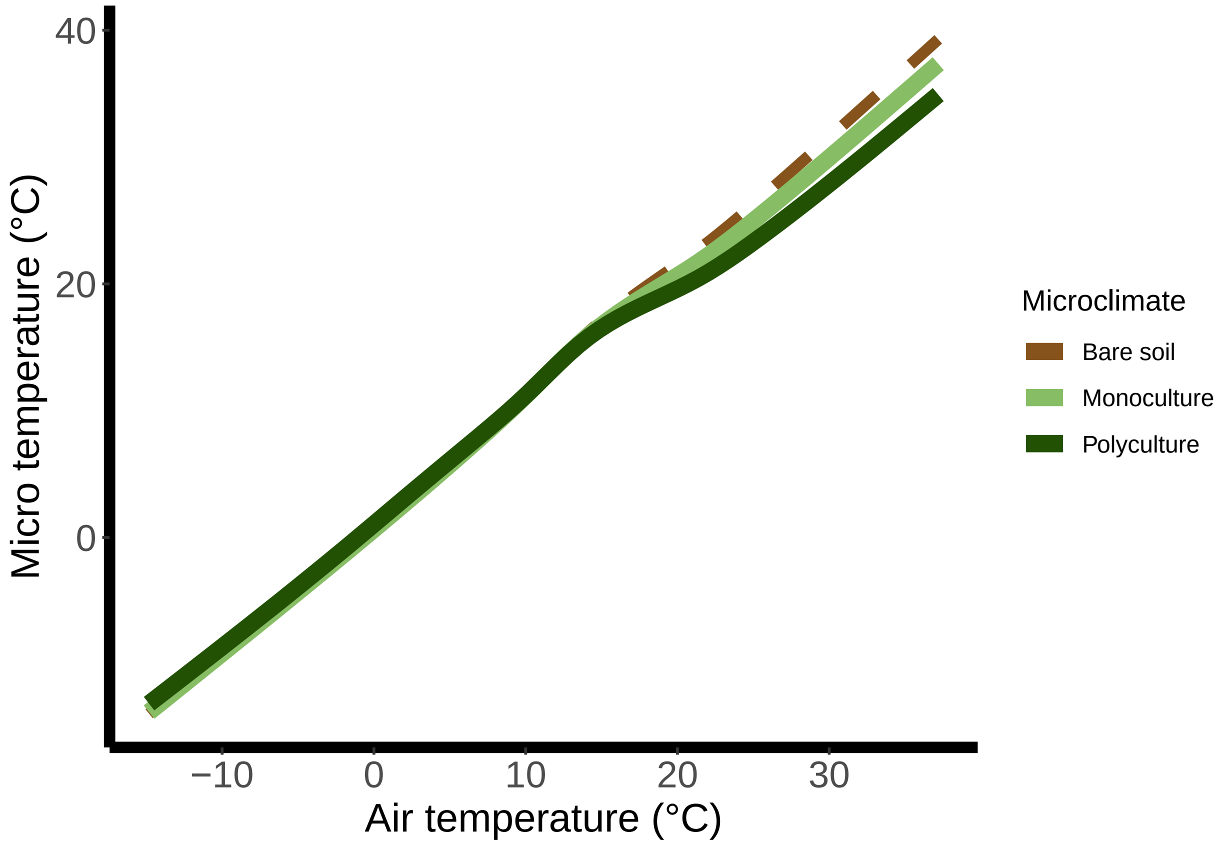
The macroclimate in function of time

```r
macroclimate=ggplot(data=df_resp_to_microclimate_growing|>
                      filter(grepl("2019-06-10|2019-06-11|2019-06-12|2019-06-13|2019-06-14",
                                   datetime)))+
            geom_line(aes(x=datetime, y=airT),
                      size=3, colour="black")+
  scale_x_datetime(name="Time (days)", date_breaks="2 days",
                    labels=c("0", "1", "3", "5", "6")
                     )+
  ylab("Air temperature (°C)")+
  my_theme

macroclimate
```



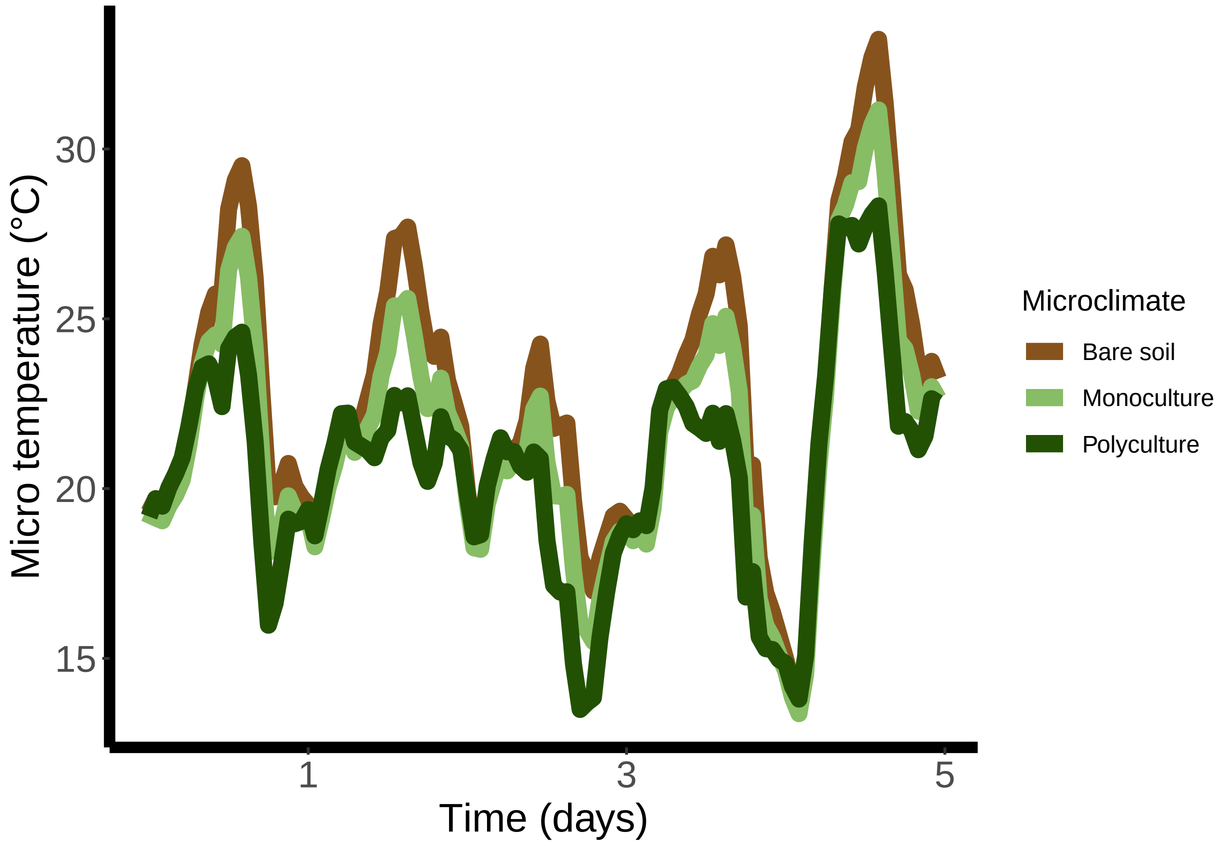
The effect of diversity on the microclimate

```
div_eff=ggplot(data=df_resp_to_microclimate_growing)+
  geom_smooth(aes(x=airT, y=soilT, colour="Bare soil"),
              se=FALSE, size=2, linetype=2)+
  geom_smooth(aes(x=airT, y=microT1, colour="Monoculture"),
              se=FALSE, size=3)+
  geom_smooth(aes(x=airT, y=microT60, colour="Polyculture"),
              se=FALSE, size=3)+
  scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                      name="Microclimate",
                      breaks=c("Bare soil", "Monoculture", "Polyculture"))+
  xlab("Air temperature (°C)")+
  ylab("Micro temperature (°C)")+
  my_theme
div_eff
```



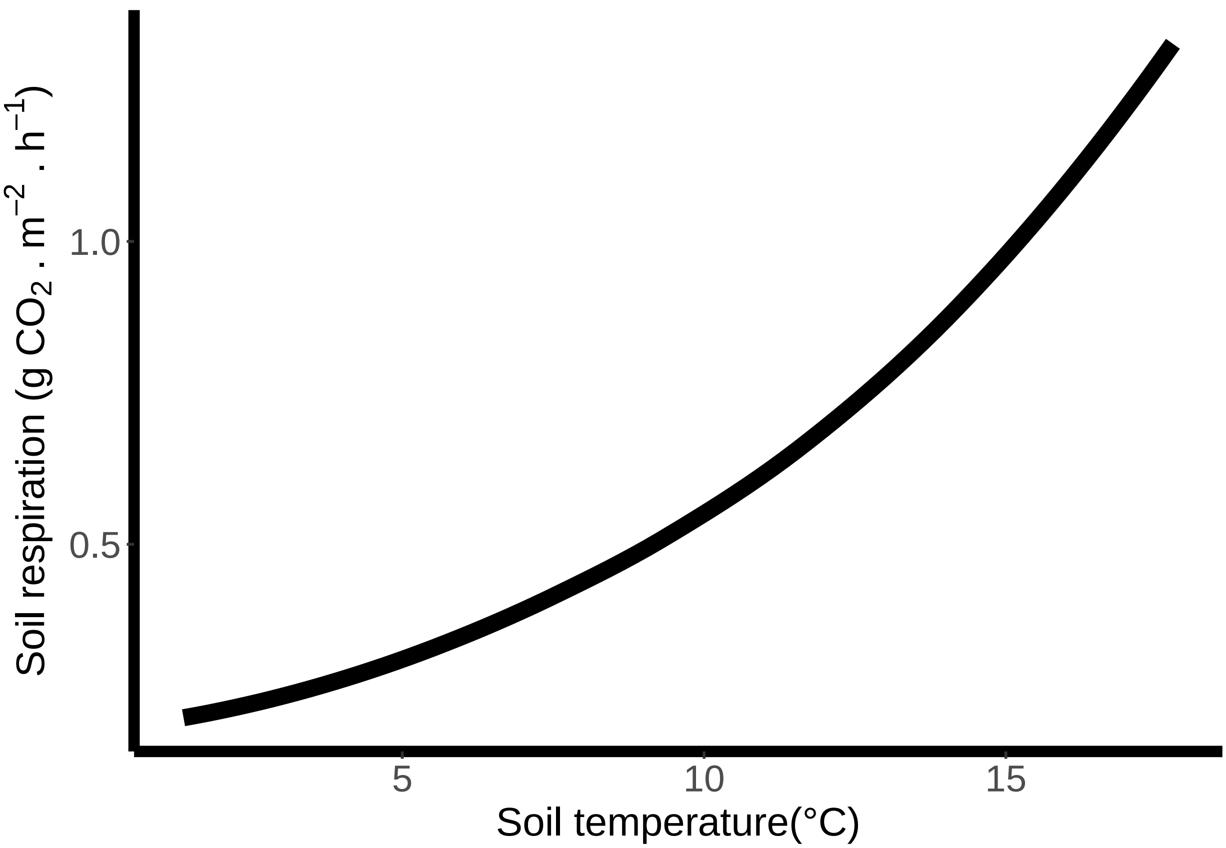
Microclimate simulation

```
micro_sim=ggplot(data=df_resp_to_microclimate_growing|>
                   filter(grepl("2019-06-10|2019-06-11|2019-06-12|2019-06-13|2019-06-14",
                                datetime)))+
  geom_line(aes(x=datetime, y=soilT, colour="Bare soil"), size=3)+
  geom_line(aes(x=datetime, y=microT1, colour="Monoculture"), size=3)+
  geom_line(aes(x=datetime, y=microT60, colour="Polyculture"), size=3)+
    scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                      name="Microclimate",
                      breaks=c("Bare soil", "Monoculture", "Polyculture"))+
   scale_x_datetime(name="Time (days)", date_breaks="2 days",
                    labels=c("0", "1", "3", "5", "6")
                     )+
  ylab("Micro temperature (°C)")+
  my_theme
micro_sim
```



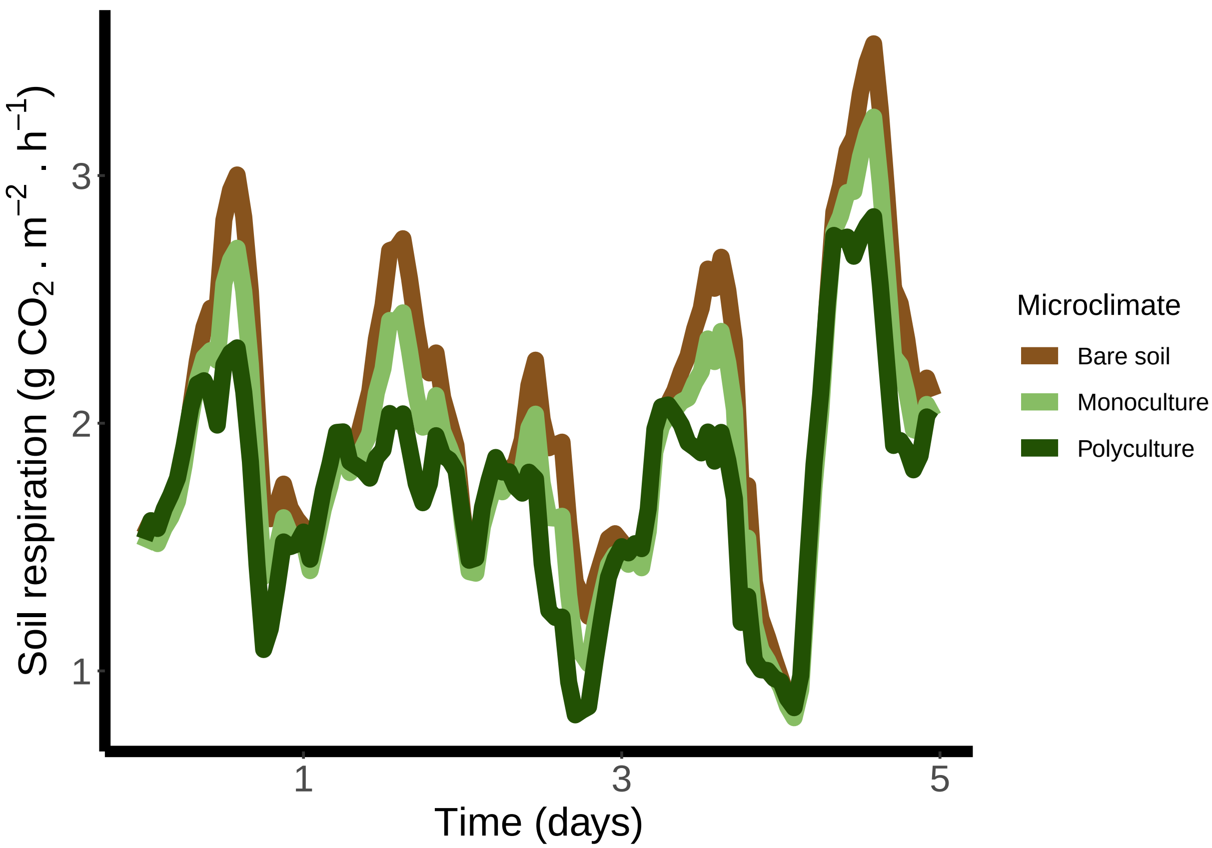
Soil respiration in function of the temperature

```
clim_dep_f=ggplot(data=tab_resp_temp)+
  geom_smooth(aes(x=temperature, y=soilresp), se=FALSE, size=3, colour="black")+
  xlab("Soil temperature(°C)")+
  ylab(bquote('Soil respiration (g'*~CO[2]~ '.'~m^-2~ '.' ~h^-1*')'))+
  my_theme
clim_dep_f
```



Ecosystem functioning: soil respiration over time

```
ecos_funct=ggplot(data=df_resp_to_microclimate_growing|>
                    filter(grepl("2019-06-10|2019-06-11|2019-06-12|2019-06-13|2019-06-14",
                                 datetime)))+
  geom_line(aes(x=datetime, y=soilrespsoilT, colour="Bare soil"), size=3)+
  geom_line(aes(x=datetime, y=soilrespmicroT1, colour="Monoculture"), size=3)+
  geom_line(aes(x=datetime, y=soilrespmicroT60, colour="Polyculture"), size=3)+
  scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                      name="Microclimate",
                      breaks=c("Bare soil", "Monoculture", "Polyculture"))+
   scale_x_datetime(name="Time (days)", date_breaks="2 days",
                    labels=c("0", "1", "3", "5", "6")
                     )+
  ylab(bquote('Soil respiration (g'*~CO[2]~ '.'~m^-2~ '.' ~h^-1*')'))+
  my_theme
ecos_funct
```



Soil respiration across time scale in function of diversity: temporal an cumulative functions (Figure 3)

```
my_theme2<- theme_classic()+theme(axis.line.x = element_line(size=1),
                                  axis.line.y=element_line(size=1),
        axis.text.x = element_text(size=7.5), axis.text.y=element_text(size=7.5),
        axis.title.x = element_text(size=9), axis.title.y=element_text(size=9))
```

Time series of temperature

```
 #Hours scale
soilRh=ggplot(data = df_resp_to_microclimate_growing|>filter(grepl("2017-04-01",
                                                                datetime))) +
  geom_smooth( aes(x = hour(datetime), y = soilrespsoilT, colour="Bare soil"),
               se=FALSE, size=2) +
  geom_smooth(aes(x = hour(datetime), y = soilrespmicroT1, colour="Monoculture"),
              se=FALSE, size=2) +
  geom_smooth(aes(x = hour(datetime), y = soilrespmicroT60, colour="Polyculture"),
              se=FALSE, size=2) +
   ylab(bquote(atop("Soil respiration",'(g'*~CO[2]*'.'*~m^-2*'.' *~h^-1*')')))+
  scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                       name="Microclimate",
                     breaks=c("Bare soil", "Monoculture", "Polyculture"))+
  scale_x_continuous(breaks = c(1,6,12,18,24))+
  scale_y_continuous(limits=c(0, 1.5))+
```

```r
  my_theme2+
  theme(axis.title.x=element_blank(),
        legend.position = "none"

)


# Months scale

soilRd=ggplot(data = df_resp_month) +
  geom_line( aes(x = mt, y = mmaT, colour="Bare soil"), size=2) +
  geom_line(aes(x = mt, y = mmiT1, colour="Monoculture"), size=2) +
  geom_line(aes(x = mt, y = mmiT60, colour="Polyculture"), size=2)+
   scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                       name="Microclimate",
                       breaks=c("Bare soil", "Monoculture", "Polyculture"))+
  scale_x_continuous(breaks = c(1, 6.0, 12.0), labels=c("Apr.", "Oct.", "Mar."))+
  scale_y_continuous(limits=c(0,2.2))+
  my_theme2+
  theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        legend.position = "none"
)

# Years scale

soilRy=ggplot(data = df_resp_year) +
  geom_line( aes(x = datetime, y = mmaT, colour="Bare soil"), size=2) +
  geom_line(aes(x = datetime, y = mmiT1, colour="Monoculture"), size=2) +
  geom_line(aes(x = datetime, y = mmiT60, colour="Polyculture"), size=2)+
  scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                      name="Microclimate",
                      breaks=c("Bare soil", "Monoculture", "Polyculture"))+
  scale_y_continuous(limits=c(0, 2.5))+
  expand_limits(x = as.Date(c("2017-04-01", "2019-12-31")))+
  my_theme2+
  theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        legend.position = "none"
)
```

Cumulative soil respiration over time

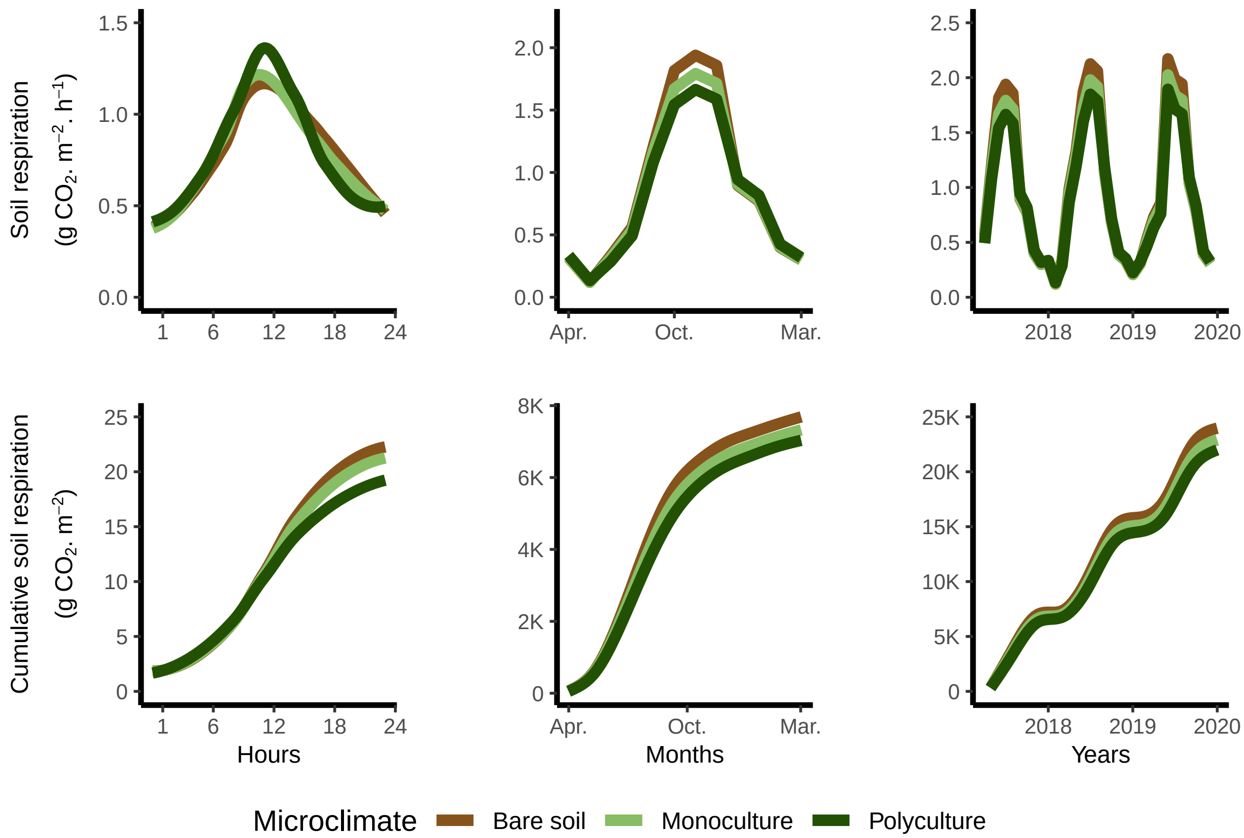
```r
#Hours scale

soilCh=ggplot(data=df_resp_to_microclimate_growing|>filter(grepl("2017-04-01",
                                                              datetime)))+
  geom_smooth(aes(x=hour(datetime), y=cumsoilrespsoilT, colour="Bare soil"),
              se=FALSE, size=2)+
  geom_smooth(aes(x=hour(datetime), y=cumsoilrespmicroT1, colour="Monoculture"),
              se=FALSE, size=2)+
  geom_smooth(aes(x=hour(datetime), y=cumsoilrespmicroT60, colour="Polyculture"),
              se=FALSE, size=2) +
```

```
  ylab(bquote(atop("Cumulative soil respiration",'(g'*~CO[2]*'.'*~m^-2*')')))+
  xlab("Hours")+
  scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                      name="Microclimate",
                     breaks=c("Bare soil", "Monoculture", "Polyculture"))+
  scale_x_continuous(breaks = c(1,6,12,18,24))+
  scale_y_continuous(limits=c(0,25),
                     breaks=c(0, 5, 10, 15, 20,25),
                     labels=c(0,5,10,15,20,25))+
  my_theme2+
  theme(
        legend.position = "none"
)


#Months scale

soilCd=ggplot(data=df_resp_to_microclimate_growing|>
  filter(year=="2017"|
           grepl("2018-03", datetime)|
           grepl("2018-02", datetime)|
           grepl("2018-01", datetime)))+
  geom_smooth(aes(x=datetime, y=cumsoilrespsoilT, colour="Bare soil"),
              se=FALSE, size=2)+
  geom_smooth(aes(x=datetime, y=cumsoilrespmicroT1, colour="Monoculture"),
              se=FALSE, size=2)+
  geom_smooth(aes(x=datetime, y=cumsoilrespmicroT60, colour="Polyculture"),
              se=FALSE, size=2) +
   scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                       name="Microclimate",
                       breaks=c("Bare soil", "Monoculture", "Polyculture"))+
  scale_x_continuous(breaks = ymd_hms("2017-04-01 01:00:00",
                                      "2017-10-04 01:00:00",
                                      "2018-03-31 01:00:00"),
                     labels=c("Apr.", "Oct.", "Mar."), name="Months")+
  scale_y_continuous(labels = scales::label_number_si())+
  my_theme2+
  theme(
        axis.title.y=element_blank(),
        legend.position = "none")


#Years scale

soilCy=ggplot(data=df_resp_to_microclimate_growing)+
  geom_smooth(aes(x=datetime, y=cumsoilrespsoilT, colour="Bare soil"),
              se=FALSE, size=2)+
  geom_smooth(aes(x=datetime, y=cumsoilrespmicroT1, colour="Monoculture"),
              se=FALSE, size=2)+
  geom_smooth(aes(x=datetime, y=cumsoilrespmicroT60, colour="Polyculture"),
              se=FALSE, size=2) +
  scale_colour_manual(values=c("#87531D", "#87BD64", "#225104"),
                      name="Microclimate",
```

```
                            breaks=c("Bare soil", "Monoculture", "Polyculture"))+
 xlab("Years")+
  scale_y_continuous(labels = scales::label_number_si(), limits=c(0,25000))+
my_theme2+
  theme(
        axis.title.y=element_blank(),
        legend.position = "none")
```

The 6 plots combined

```
t_c_f=ggarrange(soilRh, soilRd, soilRy, soilCh, soilCd, soilCy,
                nrow=2, ncol=3, align="hv", common.legend = TRUE,
                legend="bottom")
t_c_f
```



References

Copernicus Climate Change Service. n.d. "ERA5-Land Hourly Data from 2001 to Present." https://doi.org/10.24381/CDS.E2161BAC.

Huang, Yuanyuan, Gideon Stein, Olaf Kolle, Karl Kübler, Ernst-Detlef Schulze, Hui Dong, David Eichenberg, et al. 2023. "Plant Diversity Stabilizes Soil Temperature," March. https://www.biorxiv.org/content/10.1101/2023.03.13.532451v1.

Jones, S. K., R. M. Rees, D. Kosmas, B. C. Ball, and U. M. Skiba. 2006. "Carbon Sequestration in a Temperate Grassland; Management and Climatic Controls." *Soil Use and Management* 22 (2): 132–42. https://doi.org/10.1111/j.1475-2743.2006.00036.x.