Infection

Mladen Cucak

27 June, 2019

source(here::here("scr", "pkg.R"))

## [1] "All packages were successfully loaded."

BlightR <- function(weather,  
 param = NULL,  
 temporal\_res = "daily" #resoulution of the final data to be returned, daily or hourly. if hourly returned at noon  
) {  
 #' Irish Rules  
 #'  
 #' This function calculates potatolate blight risk using Irish Rules model (Bourke, 1953)  
 #' @param weather The weather data in formated as data frame.   
 #' @param infill\_gap Maximum alowed gap for missing value interpolation  
 #' @keywords Irish Rules  
  
 #Load packages   
 pkg <- c("tidyverse","lubridate")  
 if(any(pkg%in% installed.packages())==FALSE ){  
 print("Required packages need to be downloaded!")  
 } else {  
 lapply(pkg, require, character.only = TRUE)  
 }  
  
   
 source(here::here("scr", "model", "fun"))#Model functions  
  
 parameters <-   
 lapply(list.files(here::here("scr", "model", "par"),full.names = TRUE), read\_csv ) %>%   
 bind\_cols()  
   
 colnames <- c("date","doy","temp", "rhum")  
 if(all(colnames %in% colnames(fun\_df))==FALSE) stop("Rename column names. ")  
   
 if(!"hour"%in% colnames(fun\_df)) fun\_df$hour <- lubridate::hour(fun\_df$date)  
 if(!is.POSIXt(fun\_df$date)) stop("The date column is not in POSIX.. format!")  
  
   
   
 ################################  
 #Daytime calcualtions  
 #################################  
 #Calculate initiation/termiantion points for sporulation and infection  
 # Calculate an approximate sunrise and sunset times   
 #https://www.timeanddate.com/sun/@2963597  
   
 fun\_df$daytime <- NA  
   
 # Set the sunset and sunrise manually at 20/6  
 # fun\_df[fun\_df$hour == 20, "daytime"] <- "sunset"  
 # fun\_df[fun\_df$hour == 6, "daytime"] <- "sunrise"  
   
 if(!"lon" %in% colnames(fun\_df)){ stop("No Longitude reference or it is not named: 'lon'!")}  
 if(!"lat" %in% colnames(fun\_df)){ stop("No Latitude reference or it is not named: 'lat'!")}  
 lat <- fun\_df$lat  
 lon <- fun\_df$lon   
   
 #Different calculation  
 tempdf <-   
 data.frame(  
 doy = unique(fun\_df$short\_date),  
 lat = unique(lat),  
 lon = unique(lat)  
 )   
 lss <- list()  
   
   
 for (i in seq\_along(1:nrow(tempdf))) {  
 lss[[i]] <- SunInfo(tempdf[i,1], tempdf[i, 2], tempdf[i,3],1)  
 }  
   
 tempdf <-   
 lss %>% bind\_rows() %>%   
 bind\_cols(tempdf, .) %>%   
 mutate(sunset\_hr = lubridate::hour(sunset),  
 sunrise\_hr = lubridate::hour(sunrise)) %>%   
 select(c("doy", "sunrise\_hr", "sunset\_hr"))  
   
 tempdf$doy <- lubridate::yday(tempdf$doy)  
   
   
 #Define number of days for risk estimation  
 # Risk can be calculated for from day 2, because sporulaiton is calculated for   
 # the previous night. Same goes for the infection - it extends to the following morning  
 begin <- unique(fun\_df$doy)[2]   
 end <- unique(fun\_df$doy)[length(unique(fun\_df$doy))-1]   
   
 lss <- list()  
 #Define the start of sporulation   
 # i = begin  
 for(i in c(begin:end)){  
 sporstart <- tempdf[ tempdf$doy == c(i-1), "sunset\_hr"] - hr\_before\_spor  
 infstop <- tempdf[ tempdf$doy == c(i+1), "sunrise\_hr"] + hr\_after\_inf  
   
 daydf <-   
 do.call("rbind",   
 list(  
 fun\_df[with(fun\_df, hour >= sporstart & doy == c(i-1)),],# from the afternoon the day before  
 fun\_df[with(fun\_df, doy == i),],  
 fun\_df[with(fun\_df, hour < infstop & doy == c(i+1)),] #ending in the morningthe day after  
 )  
 )  
   
 ############################  
 #Sporulation  
 ############################  
 #Calculate sporulation  
 daydf$spor <- Sporulation(daydf$temp, daydf$rhum)  
   
 # Stop the sporulation n(hr\_after\_spor) hours after sunrise  
 sporstop <- tempdf[tempdf$doy == i, "sunrise\_hr"] + hr\_after\_spor   
 daydf[daydf$doy == i & daydf$hour > sporstop | daydf$doy == c(i+1), c("spor", "spor\_sum")] <- 0  
   
 #cumulative sporulation per event  
 daydf$spor\_sum <- ave(coalesce(daydf$spor, 0), data.table::rleid(zoo::na.locf(daydf$spor != 0,maxgap = 3)), FUN = cumsum)  
   
 # check if there is 10 hours for sporulation  
 criteria<- as.numeric(daydf$spor>0)  
   
 #cumulative sum of hours that meet the criteria with restart at zero  
 criteria\_sum <- stats::ave(criteria, cumsum(criteria == 0), FUN = cumsum)  
   
 risk <- rep(0, nrow(daydf))  
   
 criteria\_met <-as.numeric( criteria\_sum >= hours ) #accumulaition of EBH starts on 10th hour  
 idx <-which(criteria\_sum == hours)  
   
 # If the sporulation criteria was not met  
 if(sum(criteria\_met)==0){  
 final <- data.frame(  
 doy = i,  
 spor = max(daydf$spor\_sum, na.rm = TRUE),  
 inf = 0,  
 inf\_sol = ifelse("cumul\_inf\_sol" %in% colnames(daydf),  
 0,  
 NA )  
 )  
 lss [[i]]<-final  
   
 }else {  
 ############################  
 #Infection  
 ############################  
  
 # Calculate the infection period   
 # Starts after the conditions for sporulation have been met  
 # Sum of Infection and sporulation for each hour reduced by survival  
 daydf$inf <- 0  
 daydf[idx:nrow(daydf),"inf"] <- Infection(daydf[idx:nrow(daydf),"temp"], daydf[idx:nrow(daydf),"rhum"])  
   
 #Cumulative sum of the infection after sporulation requirement has been met  
 #The initial value is total sporulation of the day  
 daydf[idx-1,"inf"] <- max(daydf$spor\_sum)  
 daydf[c(idx-1):nrow(daydf),"cumul\_inf"] <- cumsum( daydf[c(idx-1):nrow(daydf),"inf"])  
   
 #Cumulative sum with break if the criteria is not met for five hours!!!!!!!!!!!!  
 # ave(coalesce(daydf$inf, 0), data.table::rleid(zoo::na.locf(daydf$inf != 0,maxgap = 4)), FUN = cumsum)  
   
   
 ############################  
 #Survival  
 ############################  
 # Calculate the mortalyty of spores due to solar raditaion if there is solar radiation data  
 if (mean(is.na(daydf$sol\_rad)) == 0){  
   
 #Airborne sporangia survival  
 #The estimated probablity of spore survival is calculated using   
 # The spore load was calculated as a product of total daily sporulation risk and the   
 # probability of spornagia survival asa function of solar radiation  
 Survival <- function(x) {  
 pr <- 1 / (1 + exp(-(surv\_B0 - surv\_B1 \* x)))  
 pr  
 }  
 surv\_prob <- Survival(sum(daydf[daydf$doy==i,"sol\_rad"]))  
 #Reduce the sporulation estimation based on the mortality due to solar radiation   
 reduced\_spore\_load <- max(daydf$spor\_sum)\* surv\_prob  
   
 sol\_inf<- daydf[,"inf"]  
 sol\_inf[idx-1] <- reduced\_spore\_load  
   
 daydf[,"cumul\_inf\_sol"] <- cumsum( sol\_inf) #cumulative sum of inf reduced by sol survival  
   
 }  
   
 ############################  
 #RH and tem survival !!!!!!!! SORT OUT  
 ############################  
   
 # daydf[idx:nrow(daydf),"sur"] <- Survival(daydf[idx:nrow(daydf),"temp"], daydf[idx:nrow(daydf),"rhum"])  
 # daydf[idx:nrow(daydf),"cumul\_sur"] <- cumsum( daydf[idx:nrow(daydf),"sur"])  
   
   
 # ggplot(daydf)+  
 # geom\_point(aes(date, spor\_sum, color = "blue"))+  
 # geom\_line(aes(date, spor\_sum, color = "blue"))+  
 # geom\_point(aes(date, cumul\_inf, color = "red"))+  
 #   
 # geom\_line(aes(date, cumul\_inf, color = "red"))+  
 # geom\_line(aes(date, cumul\_sur, color = "green"))+  
 # labs(title = "Daily model outputs",  
 # y = "Risk",  
 # x = "Time")+  
 # theme\_bw()  
   
 final <- data.frame(  
 doy = i,  
 spor = max(daydf$spor\_sum, na.rm = TRUE),  
 inf = max(daydf$cumul\_inf, na.rm = TRUE),  
 inf\_sol = ifelse("cumul\_inf\_sol" %in% colnames(daydf),max(daydf$cumul\_inf\_sol, na.rm = TRUE), NA )  
 )  
   
 lss [[i]]<-final  
 }  
 }  
 fin <-   
 left\_join( tempdf, bind\_rows(lss),by = "doy") %>%   
 select(-c("sunrise\_hr", "sunset\_hr"))  
   
   
 DailyAt <- function(x, time) {   
 xx <- c(sapply(x, function(x) c(rep(NA,23),x)))  
 x <- c(xx[time:length(xx)], xx[2:time])  
 return(x)  
 }  
   
   
   
   
 if(temporal\_res == "daily") {final <- fin}  
 if(temporal\_res == "hourly"){  
 fin\_hourly <-   
 data.frame(  
 spor = DailyAt(fin$spor, 18),  
 inf = DailyAt(fin$inf, 12),  
 inf\_sol = DailyAt(fin$inf\_sol, 6)  
 )  
 final <- fin\_hourly  
 }  
 return(final)  
   
  
   
 }