create.fw.list <- function(db, folder = NULL, type = NULL, ecosyst=FALSE, ref=FALSE, spatial=FALSE, code=FALSE)

{

#### Arguments ####

#'db' - database - eb (EcoBase), gw (GlobalWeb), wl (Web of Life) and mg (Mangal)

#'folder' - folder in the WD to get the dataset files (db=gw and wl). #'type' - if db=mg the user should provide the type of interactions to be

downloaded

# 'ecosyst' - Getting ecosystem information, only for gw, eb #'ref' references information

#'spatial' - get spatial info, only for wl, eb and mg

#### Results #### fwlist <- list()

#### Conditions for data entry ############################## ##############################

#db

if(!db %in% c("eb","wl","gw","mg"))stop("Argument 'db' must take one of the following values:\n

'wl' - Web of Life 'mg' - mangal 'gw' - globalweb 'eb' - ecobase")

#folder

if(!db %in% c("wl","gw") & !is.null(folder)) stop("Argument 'folder'can only be used if 'db'= 'wl' or 'gw'!")

#type

#folder

if(!db %in% c("mg") & !is.null(type)) stop("Argument 'type'can only be used if 'db'= 'mg'!")

#ecosyst

if(!db %in% c("gw", "eb") & ecosyst==TRUE) stop("Argument 'ecosyst'can only be used if 'db'= 'eb' or 'gw'!")

#ref #all

#spatial

if(!db %in% c("wl", "mg", "eb") & spatial==TRUE) stop("Argument 'spatial'can only be used if 'db'= 'eb', 'mg' 'wl'!")

#code

if(!db %in% c("wl", "mg", "gw") & code==TRUE) stop("Argument 'code'can only be used if 'db'= 'wl', 'mg'', 'gw'!")

############################################################ ################################

#Updating each dataset database

#GlobalWeb

if (db == "gw"){

message("####################### GLOBALWEB DATABASE #######################\n\n")

message("Fetching info from the provided folder!") files\_gw <- list.files(path = folder, pattern = "WEB") ngw <- length(files\_gw)

message (paste0("There are ", ngw, " food web files in the folder!"))

#Load files into list

#And create vector of references if (ref==TRUE) reflist\_gw <- c()

names\_gw <- c()#FW names

#getting the files into R for(i in 1:ngw){

message(paste0("Fetching food web ", i, " in ", ngw, "!"))

dfgw <- read.csv(paste0(folder,"/",files\_gw[i]), header = FALSE) # read csv file

dfgw <- dfgw[, colSums( is.na(dfgw) ) <=1]#Remove columns with all NA

#Get the FW name

names\_gw[i] <- as.character(dfgw[2,1])

#Get the reference to the vector

if (ref==TRUE) reflist\_gw[i] <- as.character(dfgw[1,1])

#to name the columns names\_gw\_c <- c() n1 <- ncol(dfgw)-1 for(j in 1:n1){

names\_gw\_c[j] <- as.character(dfgw[2,j+1])

}

#to name the rows names\_gw\_r <- c() n2 <- nrow(dfgw)-2 for(j in 1:n2){

names\_gw\_r[j] <- as.character(dfgw[j+2, 1])

}

dfgw <- dfgw[-c(1,2),-1]

#Remove columns with NA dfgw[dfgw==""] <- NA dfgw <- na.omit(dfgw)

if(i==281){names\_gw\_r <- names\_gw\_r[-c(36,37)]}#the FW on i=281 has a note at the bottom

#Delete the 'empty names'

names\_gw\_c <- names\_gw\_c[names\_gw\_c!=""] names\_gw\_r <- names\_gw\_r[names\_gw\_r!=""]

#Same names in rows or columns?

#if(length(unique(names\_gw\_r)) < length(names\_gw\_r)) rown[i] <- as.character(i)

#if(length(unique(names\_gw\_c)) < length(names\_gw\_c)) coln[i] <- as.character(i)

#For some strange reason some rows and columns have the same name names\_gw\_c <- paste0("sp\_", as.character(1:length(names\_gw\_c)),

"\_",names\_gw\_c)

names\_gw\_r <- paste0("sp\_", as.character(1:length(names\_gw\_r)), "\_",names\_gw\_r)

colnames(dfgw) <- names\_gw\_c rownames(dfgw) <- names\_gw\_r

fwlist[[i]] <- dfgw

}

#Name the list names(fwlist) <- names\_gw

if(ref==TRUE){

references <- as.data.frame(matrix(ncol = 4))

names(references) <- c("FW code", "first\_author", "year", "full\_ref" ) files\_gw <- list.files(folder, pattern = "WEB")

message("Fetching references from the dataset files!") for(w in 1:ngw){

dfgw <- read.csv(paste0(folder,"/",files\_gw[w]), header = FALSE) # read csv file

#message(paste0("Reading file ", files\_gw[w]))

dfgw <- dfgw[, colSums( is.na(dfgw) ) <=1]#Remove columns with all NA

#Get the reference to the vector full\_ref1 <- as.character(dfgw[1,1])

references[w,4] <- full\_ref1#full reference references[w,1] <- files\_gw[w]#fw code

references[w,2] <- str\_sub(word(full\_ref1, start = 1), 1, str\_length(word(full\_ref1, start = 1))-1)#fisrt author

references[w,3] <- regmatches(x = full\_ref1,gregexpr("[0-9]+",text = full\_ref1))[[1]][1]#year

#references[w,3] <- gsub('.+\\(([0-9]+)\\).+?$', '\\1', full\_ref1)#year

}#end loop to add refs

}#end gw refs #ECOSYSTEM

if(ecosyst==TRUE){

message("Searching for 'gw\_list.csv' file...")

if (!file.exists(paste0(folder, "/gw\_list.csv"))) stop("\nThe pdf 'gw\_list.pdf' has to be previously converted to a csv file...")

#I had to conver the gw\_list.pdf file to excel (csv), since I could not install tabulizes to extract pdf tables

gw\_eco <- read.csv(paste0(folder,"/","gw\_list.csv"), header = TRUE, sep = ";") # read csv file

filn <- paste0("WEB", as.character(gw\_eco[,1]), ".csv")

gw\_eco2 <- gw\_eco[,1:3] gw\_eco2[,1] <- filn

names(gw\_eco2)[1] <- "FW"

#yes... I do know the following few lines are 'ugly'... filn <- as.data.frame(cbind(filn, filn))

names(filn) <- c("filn1","filn2")

#files\_gw <- list.files(path = folder, pattern = "WEB") ecosystem <- merge(x=filn, y=gw\_eco2, by.x= "filn2", by.y = "FW") ecosystem <- ecosystem[,c(2, 3, 4)]

names(ecosystem)[1] <- "Food web"

}

}#end of gw #Web of Life

if (db == "wl"){

message("####################### WEB OF LIFE DATABASE #######################\n\n")

files\_wl <- list.files(path = folder, pattern = "FW") nwl <- length(files\_wl)

message (paste0("There are ", nwl, " food web files in the folder!"))

#Get refs and metrics table

if (file.exists(paste0(folder, "/references.csv"))) {

table\_wl <- read.csv(paste0(folder, "/references.csv"), header = TRUE) #

read csv file

} else {

stop("There is no 'references.csv' file on the folder, as provided by the website!")

}

#FW names

names\_wl <- as.character(table\_wl[,8])

#Load files for(i in 1:nwl){

message(paste0("Fetching food web ", i, " in ", nwl, "!"))

dfwl <- read.csv(paste0(folder, "/",files\_wl[i]), header = TRUE) # read csv file

#row.names(dfwl) <- as.character(dfwl[,1]) #dfwl <- dfwl[,-1]

dfwl[is.na(dfwl)] <- 0 fwlist[[i]] <- dfwl

}

names(fwlist) <- names\_wl #REFERENCES

if(ref==TRUE){

references <- as.data.frame(matrix(ncol = 4))

names(references) <- c("FW code", "first\_author", "year", "full\_ref" )

message("Fetching references from the 'references.csv' file!") message("Checking the presence of the 'references.csv' file...") if(!file.exists(paste0(folder, "/references.csv"))==TRUE)stop("Can't

retrieve reference details... \n File not present!")

ref\_file <- read.csv(paste0(folder, "/references.csv"), header = TRUE) # read csv file

for(w in 1:nwl){

full\_ref1 <- as.character(ref\_file[w,7]) references[w,4] <- full\_ref1#full reference references[w,1] <- as.character(ref\_file[w,1])#fw code

references[w,2] <- str\_sub(word(full\_ref1, start = 1), 1, str\_length(word(full\_ref1, start = 1))-1)#fisrt author

references[w,3] <- regmatches(x = full\_ref1,gregexpr("[0-9]+",text = full\_ref1))[[1]][1]#year

#references[w,3] <- gsub('.+\\(([0-9]+)\\).+?$', '\\1', full\_ref1)#year

}#end loop to add refs

}#end wl refs #SPATIAL

if(spatial==TRUE){

message("Fetching the spatial information from the 'references.csv' file!")

message("Checking the presence of the 'references.csv' file...") if(!file.exists(paste0(folder, "/references.csv"))==TRUE)stop("Can't

retrieve spatial info... \n File not present!")

ref\_file <- read.csv(paste0(folder, "/references.csv"), header = TRUE) # read csv file

spatial1 <- ref\_file[,c(1,9,10)]

}#end of spatial

}#end of wl #EcoBase

if(db == "eb"){

message("####################### ECOBASE DATABASE #######################\n\n")

message("Fetching info from the EcoBase website!") suppressWarnings({

#To obtain the list of available models

suppressMessages({ h=basicTextGatherer()

curlPerform(url = '<http://sirs.agrocampus-ouest.fr/EcoBase/php/webser/soap-> client\_3.php',writefunction=h$update)

data1 <- xmlTreeParse(h$value(),useInternalNodes=TRUE)

liste\_mod <- ldply(xmlToList(data1),data.frame)#liste\_mod contains a list and decription

})

#Select only those allowing dissemination

l2 <- subset(liste\_mod, model.dissemination\_allow =="true")#only those of which dissemination is allowed

message("Sellected only those to which model dissemination is allowed!")

#Select only those with whole food webs

l3 <- subset(l2, model.whole\_food\_web =="true")#only those with the full food web

message("Sellected only those to which the whole food web is available!")

#Get model names

model.name <- as.character(l3$model.model\_name) input\_list <- list()

id <- as.numeric(as.character(l3$model.model\_number))

#Loop to get input list for(i in 1:nrow(l3)){

message(paste0("Fetching information on food web ",i, " of ", nrow(l3)))

suppressMessages({ h=basicTextGatherer() mymodel <- id[i]

curlPerform(url = paste('<http://sirs.agrocampus-ouest.fr/EcoBase/php/webser/soap-> client.php?no\_model=',mymodel,sep=''),writefunction=h$update,verbose=TRUE)

data2 <- xmlTreeParse(h$value(),useInternalNodes=TRUE)

input1 <- xpathSApply(data2,'//group',function(x) xmlToList(x))

})

#need do name the columns

names\_input <- as.character(input1[1,]) input1 <- as.data.frame(input1) colnames(input1) <- names\_input

input1 <- input1[-1,] input\_list[[i]] <- input1

}#end of loop to get input list mnames <- names(input\_list)

for (i in 1:length(input\_list)){

m2 <- input\_list[[i]] #get the model nnodes <- length(m2)

node\_names <- names(m2)

# if (biomass == TRUE) # {

# nodes\_biomass <- as.data.frame(matrix(ncol=3, nrow=nnodes)) # names(nodes\_biomass) <- c("id", "name", "biomass")

# }

int\_matrix <- as.data.frame(matrix(ncol=nnodes, nrow=nnodes)) for(j in 1:length(m2)){

node1 <- m2[[j]]

node\_id <- as.numeric(node1$group\_seq) #node1\_biomass <- as.numeric(node1$biomass) node\_name <- node\_names[j]

#biomass

#if (biomass == TRUE) #{

#nodes\_biomass[node\_id, 1] <- node\_id #nodes\_biomass[node\_id, 2] <- node\_name #nodes\_biomass[node\_id, 3] <- node1\_biomass #}

#matrix

colnames(int\_matrix)[node\_id] <- node\_name rownames(int\_matrix)[node\_id] <- node\_name

diet\_node1 <- node1$diet\_descr nr\_food\_items <- length(diet\_node1)

for(a in 1:nr\_food\_items){ item1 <- diet\_node1[[a]]

id\_item1 <- as.numeric(item1$prey\_seq) proportion\_item1 <- as.numeric(item1$proportion) detritus\_item1 <- as.numeric(item1$detritus\_fate) #send to matrix

int\_matrix[id\_item1,node\_id] <- proportion\_item1

}

}

int\_matrix[is.na(int\_matrix)] <- 0#replacing NA with 0 #if(db=="eb" && biomass == TRUE) fwlist[[i]] <-

list(biomass=nodes\_biomass, trophic\_relations=int\_matrix) #if(db=="eb" && biomass == FALSE) fwlist[[i]] <- int\_matrix fwlist[[i]] <- int\_matrix

}

names(fwlist) <- model.name

})#end of outer suppressWarnings #REFERENCES

if(ref==TRUE){

references <- as.data.frame(matrix(ncol = 4))

names(references) <- c("FW code", "first\_author", "year", "full\_ref" ) message("Fetching the references information!")

for(w in 1:nrow(l3)){

code

#Get the reference to the vector

full\_ref1 <- as.character(l3$model.reference)[w] references[w,4] <- full\_ref1#full reference

references[w,1] <- as.numeric(as.character(l3$model.model\_number[w]))#fw

references[w,2] <- as.character(l3$model.author[w])#fisrt author references[w,3] <- regmatches(x = full\_ref1,gregexpr("[0-9]+",text =

full\_ref1))[[1]][1]#year

#references[w,3] <- gsub('.+\\(([0-9]+)\\).+?$', '\\1', full\_ref1)#year

}#end loop to add refs

}#end of eb refs #ECOSYSTEM

if(ecosyst==TRUE){

ecosystem <- data.frame(l3$model.model\_number, l3$model.country, l3$model.ecosystem\_type)

names(ecosystem) <- c("Food web", "Location", "Ecosystem")

}#end of eb ecosystem #SPATIAL

if(spatial==TRUE){

message("Fetching spatial information from the EcoBase website...")

#Get actual polygons

EcoBase\_shape <- sf::st\_read("<http://sirs.agrocampus-ouest.fr/EcoBase/php/protect>

/extract\_kml.php")

ebd <- EcoBase\_shape$Name

#Getting the model numbers nmr <- list()

for(i in 1:length(ebd)){

nr <- strsplit(as.character(ebd[i]), "--::")[[1]][1]

nr <- as.numeric(str\_extract\_all(nr, "\\d+")[[1]])#Alternative to Numextract

nmr[[i]] <- nr

}

nmr2 <- c()#line rows for each model for(i in 1:length(nmr)){

a <- nmr[[i]] b <- length(a) c1 <- rep(i,b)

nmr2 <- c(nmr2, c1)

}

#In Which row in ecobase geo file is the model? nmr <- unlist(nmr)

table1 <- as.data.frame(cbind(nmr2, nmr)) colnames(table1) <- c("row\_n","id")

#In which row does model.model\_number with a given Id occurs? lines\_n <- c()

for (i in 1:nrow(liste\_mod)){

id <- as.numeric(as.character(liste\_mod$model.model\_number[i])) lines\_n[i] <- as.numeric(table1[table1$id==id,][1])

}

ecobase\_poly2 <- list() for(i in 1:length(lines\_n)){

ecobase\_poly2[i] <- st\_geometry(EcoBase\_shape)[lines\_n[i]] #plot(st\_geometry(EcoBase\_shape)[lines\_n[i]], border="green", add=TRUE)

}

#if no polygon then bounding box #into here ecobase\_poly2

for(i in 1:length(ecobase\_poly2)){ if(is.na(lines\_n[i])){

#create a bounding box geographic thing

z1 <- as.numeric(Numextract(liste\_mod$model.geographic\_extent[[i]])) z2 <- c(z1[4], z1[1], z1[2], z1[1], z1[2], z1[3], z1[4], z1[3])

x1 <- as.data.frame(matrix(z2, ncol=2, byrow=TRUE))

x1 <- cbind(x1[2], x1[1])#had to change lat and long... I had this the other way around...

p1 <- Polygon(x1)

ps1 <- Polygons(list(p1),1)

ecobase\_poly2[[i]] <- st\_as\_sf(SpatialPolygons(list(ps1)))

}

ecobase\_poly2[[i]] <- ecobase\_poly2[[i]]

}

#convert all to class sf

for(i in 1:length(ecobase\_poly2)){ if(!any(class(ecobase\_poly2[[i]])=='sf')){

t2 <- ecobase\_poly2[[i]]

t3 <- st\_cast(t2, to="POLYGON")

ecobase\_poly2[[i]] <- st\_as\_sf(as(st\_zm(st\_geometry(t3)), "Spatial"))

}

else message("Ok!")

}

#line.Id correspondence

table2 <- as.data.frame(cbind(1:length(ecobase\_poly2),as.numeric(as. character(liste\_mod$model.model\_number))))

names(table2) <- c("row","id")

#select the corresponding polygons

id\_selected <- as.numeric(as.character(l3$model.model\_number))

#Which rows? rows\_selected <- c()

for(i in 1:length(id\_selected)){

rows\_selected[i] <- as.numeric(table2[table2["id"]

==id\_selected[i],][1])

}

spatial1 <- ecobase\_poly2[rows\_selected]

}#end of eb spatial

}#end of eb #MANGAL

if(db == "mg"){

message("####################### MANGAL DATABASE #######################\n\n")

message("Fetching datasets from the Mangal website! \n\n Types 'predation' and 'herbivory' by default... \n but run mangal function 'avail\_type' to check available types...\n\nThis operation might take a long time!")

ntypes <- length(type) net\_info <- list() for(i in 1:ntypes){

message(paste0("\n\nFetching information from interactions of the type ","'",type[i], "'!"))

fwlist1 <- search\_interactions(type = type[i]) %>% get\_collection() net\_info <- rbind(net\_info, fwlist1)

fwlist2 <- as.igraph(fwlist1)

fwlist <- c(fwlist, fwlist2) #class(fwlist)

}

#Converting igraph objects to data frame for(i in 1:length(fwlist)){

fw2 <- fwlist[[i]]

#convert each igraph to a data frame fw3 <- as\_data\_frame(fw2, what = "both") id\_name <- fw3$vertices[,1:2]

for(j in 1:nrow(id\_name)){#clean the names node\_name <- id\_name$original\_name[j]

if (grepl(":", node\_name, fixed=TRUE)) {

node\_name <- tail(strsplit(node\_name, ": "))[[1]] id\_name[j,2] <- node\_name[2]

} else id\_name[j,2] <- node\_name

}#end clean names

id\_edges <- fw3$edges[,1:3]

int\_matrix <- as.data.frame(matrix(ncol = nrow(id\_name), nrow = nrow(id\_name)))

colnames(int\_matrix) <- id\_name$original\_name rownames(int\_matrix) <- id\_name$original\_name

#Fill the matrix

for(a in 1:nrow(id\_edges)){

edge1 <- as.numeric(id\_edges[a,1:2])

name1 <- id\_name[as.character(edge1[1]),][,2] name2 <- id\_name[as.character(edge1[2]),][,2] int\_matrix[name1,name2] <- 1

}

int\_matrix[is.na(int\_matrix)] <- 0 #convert all NA to zero fwlist[[i]] <- int\_matrix

}#end of loop to convert to a data frame

if(ref==TRUE){

references <- as.data.frame(matrix(ncol = 4))

names(references) <- c("Dataset ID", "first\_author", "year", "DOI" ) message("Fetching references!")

for(j in 1:length(net\_info)){

dataset\_id <- net\_info[[j]]$dataset$dataset\_id first\_author <- net\_info[[j]]$reference$first\_author year\_mng <- as.numeric(net\_info[[j]]$reference$year) doi\_mng <- net\_info[[j]]$reference$doi

references[j,1] <- dataset\_id references[j,2] <- first\_author references[j,3] <- year\_mng references[j,4] <- doi\_mng

references <- references[order(references$`Dataset ID`),] rownames(references) <- 1:nrow(references)

}

}#End of mg refs if(spatial==TRUE){

spatial1 <- as.data.frame(matrix(ncol = 4))

names(spatial1) <- c("Dataset ID", "first\_author", "lat", "long") message("Fetching coordinates!")

for(z in 1: length(net\_info)){

dataset\_id <- net\_info[[z]]$dataset$dataset\_id lat\_mng <- net\_info[[z]]$network$geom\_lat long\_mng <- net\_info[[z]]$network$geom\_lon

first\_author <- net\_info[[z]]$reference$first\_author if(length(unlist(lat\_mng))>1){

spatial2 <- as.data.frame(matrix(ncol = 4))

names(spatial2) <- c("Dataset ID", "first\_author", "long", "lat" )

for(b in 1:length(unlist(lat\_mng))){ spatial2[b,3] <- long\_mng[[1]] [b]

spatial2[b,4] <- lat\_mng [[1]] [b]

}

spatial2[,1] <- dataset\_id spatial2[,2] <- first\_author

spatial1 <- rbind(spatial1, spatial2)

}

spatial1[z,1] <- dataset\_id spatial1[z,2] <- first\_author

if(length(unlist(lat\_mng))==1) spatial1[z,3] <- lat\_mng if(length(unlist(lat\_mng))==1) spatial1[z,4] <- long\_mng

}

spatial1 <- spatial1[order(spatial1$`Dataset ID`),] rownames(spatial1) <- 1:nrow(spatial1)

}#End of mg spatial

if (exists("references") & exists("spatial1")) (if(nrow(references)!=nrow( spatial1)) message("WARNING: There are more than on FW in some datasets!

References and Spatial data frames have different number of rows."))

}#end of mangal

message(paste0("DONE! \n\nOverall the list stores ", length(fwlist), " datasets!"))

master\_list <- list() master\_list[["int\_matrix"]] <- fwlist

if(ecosyst==TRUE) { master\_list[["ecosystem"]] <- ecosystem

message ("\n Additional element in the results: \n\n The vector with information on the ecosystems.")

}

if(ref==TRUE) { master\_list[["references"]] <- references

message ("Additional element in the results! \nA data frame with information on the references.")

}

if(spatial==TRUE) { master\_list[["spatial\_info"]] <- spatial1

message ("\n Additional element in the results: \n\n Spatial information was added.")

}

if(code==TRUE) {

if(db == "gw") master\_list[["code"]] <- files\_gw if(db == "wl") master\_list[["code"]] <- files\_wl if(db == "mg") master\_list[["code"]] <- references[1,]

message ("Added food web code information.")

}

#Return results if(length(master\_list)==1) return(fwlist)

if(length(master\_list)!=1) return(master\_list) message("####################### DONE! #######################")

}#END OF FUNCTION create.fw.list