Filtering_datasets.r

Usuario

2025-06-02

```
#Title: Filtering_datasets
#Created 29/5/2025
#Authors: Solimary García & Micaela Santos
#Project: NEFINEO_MS.Rproj
#Description: The aim of this code is to get the final globalfungi dataset for its1 and its2.
#1. Create a column to keep samples according to paper id (no plantations).
#2. Create a column to group samples by a threshold of 90m of proximity.
#3. Create a new permanent ID to each sample according to the number of group and ...
#the year of paper to avoid getting grouped samples of different samplings.
#4. Export completed datasets as .csv files
#5. Export short datasets indicating new ID and permanent ID of each sample

#Set working directory from NEFINEO_MS.Rproj####
#github directory
list.files() #check files in folder
```

```
## [1] "Buscando_Replicas.R"          "Data"
## [3] "Filtering_datasets.pdf"       "Filtering_datasets.r"
## [5] "Filtering_datasets.spin.R"    "Filtering_datasets.spin.Rmd"
## [7] "NEFINEO_MS.Rproj"             "README.md"
## [9] "soilgrid_WP2.R"
```

```
#Datasets in folder: Data

#Open datasets ####
#final dataset its1
its1= read.table("Data/globfungi_metadata_filtered_complemented_its1_soil.tsv",
                 sep ="\t",)
length(unique(its1$PermanentID)) #check ID by sample #2815 samples
```

```
## [1] 2815
```

```
#final dataset of articles to keep in its1 (no plantations)
its1.plant= read.csv("Data/article_list_noplantation.its1.csv")
#final dataset its2
its2= read.table("Data/globfungi_metadata_filtered_complemented_its2_soil.tsv",
                 sep ="\t",)
length(unique(its2$PermanentID)) #check ID by sample #2380 samples
```

```
## [1] 2380
```

```r
#final dataset of articles to keep in its2 (no plantations)
its2.plant= read.csv("Data/article_list_noplantation.its2.csv")

#Create new vector of articles to keep in datasets its1/its2 ####
#its1
keep.its1= subset(its1.plant, To_keep=="yes", select= title) #articles to keep
nrow(keep.its1) #20 articles to keep
```

```
## [1] 20
```

```r
v.keep.its1= match(its1$paper_id, keep.its1$title) #vector with articles to keep
v.keep.its1= ifelse(is.na(v.keep.its1), "no", "yes") #reemplace Na by "no"
#its2
keep.its2= subset(its2.plant, To_keep=="yes", select= title) #articles to keep
nrow(keep.its2) #28 articles to keep
```

```
## [1] 28
```

```r
v.keep.its2= match(its2$paper_id, keep.its2$title) #vector with articles to keep
v.keep.its2= ifelse(is.na(v.keep.its2), "no", "yes") #reemplace Na by "no"

#Add new vector of articles to keep to dataset ####
new.its1= cbind(its1[,1:5], paper_to_keep=v.keep.its1, its1[,6:171])
new.its2= cbind(its2[,1:5], paper_to_keep=v.keep.its2, its2[,6:171])

#Create new vector of grouped samples by a threshold of 90m of proximity ####
#install packages if need it
library (dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library (geosphere)
```

```
## Warning: package 'geosphere' was built under R version 4.3.3
```

```r
#matrix distance of sampling sites of its1
dist_matrix_its1 <- distm(new.its1[,c("longitude","latitude")], fun = distHaversine)
#matrix distance of sampling sites of its2
dist_matrix_its2 <- distm(new.its2[,c("longitude","latitude")], fun = distHaversine)

#Performing hierarchical clustering based on geographic distance
```

```
cluster_its1 <- hclust(as.dist(dist_matrix_its1), method = "complete")
cluster_its2 <- hclust(as.dist(dist_matrix_its2), method = "complete")

#Cutting the dendrogram into clusters using threshold of 90m
groups_its1 <- cutree(cluster_its1, h = 90) #new vector of grouped samples
groups_its2 <- cutree(cluster_its2, h = 90) #new vector of grouped samples

#Add new vector of grouped samples to datasets ####
new.its1= cbind(new.its1[,1:3], grouped_samples=groups_its1, new.its1[,4:172])
new.its2= cbind(new.its2[,1:3], grouped_samples=groups_its2, new.its2[,4:172])

#Create new permanent ID for grouped samples ####
new.ID.its1= paste0("NEF_its1_",new.its1$grouped_samples,"_",new.its1$year)
new.ID.its2= paste0("NEF_its2_",new.its2$grouped_samples,"_",new.its2$year)

#Add column for new ID
new.its1= cbind(new.ID= new.ID.its1, new.its1[,1:173])
new.its2= cbind(new.ID= new.ID.its2, new.its2[,1:173])

#Checking final datasets ####
length(which(new.its1[,"paper_to_keep"]=="yes",TRUE)) #1829 non-grouped samples
```

```
## [1] 1829
```

```
final.its1= subset(new.its1, paper_to_keep=="yes") #filtering by papers to keep
length(unique(final.its1$grouped_samples)) #563 grouped samples
```

```
## [1] 563
```

```
length(which(new.its2[,"paper_to_keep"]=="yes",TRUE))#1435 non-grouped samples
```

```
## [1] 1435
```

```
final.its2= subset(new.its2, paper_to_keep=="yes") #filtering by papers to keep
length(unique(final.its2$grouped_samples)) #729 grouped samples
```

```
## [1] 729
```

```
#check grouped samples from more than 1 study
#its1
studies.its1= tapply(final.its1$paper_id, final.its1$new.ID, function(x) length(unique(x)))
n.studies.its1= studies.its1[studies.its1 > 1]#grouped samples (new.ID) including  more than 1 study
#its2
studies.its2= tapply(final.its2$paper_id, final.its2$new.ID, function(x) length(unique(x)))
n.studies.its2= studies.its2[studies.its2 > 1]

#Export datasets as .csv ####
#write.csv(new.its1, file="Data/new.its1.csv")
#write.csv(new.its2, file="Data/new.its2.csv")
```

```
#Export datasets as .csv ####
#Filter useful columns
to_use_its1= subset(new.its1, paper_to_keep=="yes", select=c(new.ID,PermanentID))
to_use_its2= subset(new.its2, paper_to_keep=="yes", select=c(new.ID,PermanentID))

#write.csv(to_use_its1, file="Data/short.new.its1.csv")
#write.csv(to_use_its2, file="Data/short.new.its2.csv")
```