Format original data from Kolpelke et al. 2016 with R

By Jens-Peter Kopelke, Tommi Nyman, Kevin Cazelles, Dominique Gravel, Steve Vissault and Tomas Roslin

November 18, 2016

Abstract

For conciseness, the data are offered as a single spreadsheet with its variables defined in section B (above). To provide the tools for the correct import of this complex data set in R, for reshaping the spreadsheet format as a series of relational objects, and for exploring the resulting data structure, we here provide R code for the benefit of the data user. The approach is based on splitting the data set into different files for which primary keys (i.e. unique identifiers) are assigned, thus allowing the user to easily retrieve pieces of information from each files. Below, we offer examples of how to handle the data, how to obtain a quick map and how to use the data for network analyses.

1 R and add-on packages

For this purpose, we use R (version >= 3.2; R Core Team 2016) and the following set of add-on packages:

Table 1: Add-on packages used.

packages	version
bipartite	2.07
dismo	1.1-1
igraph	1.0.1
magrittr	1.5
rgdal	1.1.10
raster	2.5.8
reshape2	1.4.2
sp	1.2.3

To run the following lines of code properly, the above packages must first be installed. For instance, to function used to reshape the data set requires function from the *magrittr* package to be installed and loaded. The code below install this specific package if not available, the user can do the same for all the packages listed above.

```r
if(!require(magrittr)) install.packages(magrittr)
library(magrittr)
```

2 Reshaping the original dataset

The original dataset can be reshaped as follows:

Cleaning the existing `./csv/` and `./rdata/` folders if any,

```r
unlink("./csv", recursive = TRUE)
unlink("./rdata", recursive = TRUE)
```

Importing the R script which contains the reshaping function:

```
source("./lib/format4R.r")
```

Applying the reshaping function to the original dataset:

```
get_formatData("./Salix_webs.csv")
```

Two new folders have now been created (`./csv/` and `./rdata/`) in your working directory, within which six files have been added with the following contents:

File	row	description
df_site	2029	Location and of the sites
df_salix	52	Information on willow nodes
df_galler	96	Information on sawfly nodes
df_parasit	126	Information on parasitoids nodes
df_interact	4749	Interactions details among each node (willows, gallers, sawflies)
df_salix_galler	2029	Supplementary information on the interaction among willows and gallers

3 Exploring the new data structure

3.1 Sampling sites

The next few lines will import and display the structure of the file describing the sampling units (i.e. file ./rdata/df_site.rds).

```
df_site <- readRDS("./rdata/df_site.rds")
str(df_site, strict.width="cut")
```

```
## 'data.frame':    2029 obs. of  9 variables:
##  $ REARING_NUMBER: Factor w/ 2029 levels "198203W1-R30vimin",..: 320 321 ..
##  $ YEAR_OF_COLL  : num  1987 1987 1987 1987 1987 ...
##  $ LEG           : Date, format: "1987-07-20" "1987-07-20" ...
##  $ COUNTRY       : chr  "Austria" "Austria" "Austria" "Austria" ...
##  $ REGION        : chr  "Tirol" "Tirol" "Tirol" "Tirol" ...
##  $ SITE          : chr  "Gern-Alm" "Gern-Alm" "Zillertal, Hintertux/ Wei"..
##  $ NDECDEG       : num  47.5 47.5 47.1 47.4 47.5 ...
##  $ EDECDEG       : num  11.6 11.6 11.7 11.8 11.6 ...
##  $ ELEVATION     : num  1253 1253 1769 1875 1253 ...
```

In this file, each row refers to a willow species sampled at a specific time (`YEAR_OF COLL`) in a given location (`SITE`). The field `REARING NUMBER` is the primary key of this table, and thus points to a unique record.

3.2 Nodes

The command lines below will import and display the structure of the tables (available in `./rdata`) associated with the different levels of the network.

3.2.1 Willow species (df_salix.rds)

```
df_salix <- readRDS("./rdata/df_salix.rds")
str(df_salix, strict.width="cut")
```

```
## 'data.frame':    52 obs. of  3 variables:
##  $ RSAL   : Factor w/ 52 levels "Sal1","Sal10",..: 1 12 23 34 45 49 50 51..
##  $ SPECIES: chr  "elaeagnos" "appendiculata" "myrsinifolia" "foetida" ...
##  $ AUTHOR : chr  "Scop. 1772" "Villars 1789" "Salisbury 1796" "Schleich."..
```

This file contains information on the Salix species, with `RSAL` as its unique identifier (primary key).

3.2.2 Galler species (`df_galler.rds`)

```
df_galler <- readRDS("./rdata/df_galler.rds")
str(df_galler, strict.width="cut")
```

```
## 'data.frame':    96 obs. of  7 variables:
##  $ RGALLER                 : Factor w/ 96 levels "Eangus","Eappen",..: 63..
##  $ GENUS                   : chr  "Pontania" "Pontania" "Pontania" "Pont"..
##  $ SPECIES                 : chr  "elaeagnocola" "bridgmanii" "varia" "o"..
##  $ GENUS_SPECIES           : chr  "Pontania elaeagnocola" "Pontania brid"..
##  $ AUTHOR                  : chr  "Kopelke 1994" "(Cameron 1883)" "Kopel"..
##  $ CODE_GALLTYPE           : chr  "RK1" "RK3" "RK4" "RK3" ...
##  $ CODE_GALLTYPE SIMPLIFIED: chr  "Leaf blade sausage gall" "Leaf blade "..
```

This file contains information on galler species, with `RGALLER` as its unique identifier.

3.2.3 Parasitoid species (`df_parasit.rds`)

```
df_parasit <- readRDS("./rdata/df_parasit.rds")
str(df_parasit, strict.width="cut")
```

```
## 'data.frame':    126 obs. of  11 variables:
##  $ RPAR                      : Factor w/ 126 levels "Aacumi","Aalvea",..:..
##  $ ORDER                     : chr  "Hymenoptera" "Hymenoptera" "Hymenop"..
##  $ SUPERFAMILY               : chr  "Chalcidoidea" "Chalcidoidea" "Chalc"..
##  $ FAMILY                    : chr  "Eulophidae" "Eulophidae" "Eulophida"..
##  $ GENUS                     : chr  "Anaprostocetus" "Aprostocetus" "Apr"..
##  $ GOODNESS OF ID            : chr  "species" "species" "species" "genus"..
##  $ P/I                       : chr  "P" "P" "P" "P" ...
##  $ ENDO/ECTO                 : chr  "Endo" "Endo" "Endo" "Endo" ...
##  $ KOINO/IDIO                : chr  "Koino" "Koino" "Koino" "Koino" ...
##  $ 1INSTAR/LINSTAR/COCOON/EGG: chr  "1INSTAR" "1INSTAR" "1INSTAR" "1INST"..
##  $ FULL_NAME                 : chr  "Anaprostocetus acuminatus (Ratzebur"..
```

This file contains information on parasitoid species, with `RPAR` as its unique identifier.

3.3 Links

```
df_interact <- readRDS("./rdata/df_interact.rds")
str(df_interact, strict.width="cut")
```

```
## 'data.frame':    4749 obs. of  6 variables:
##  $ REARING_NUMBER: Factor w/ 2029 levels "198203W1-R30vimin",..: 320 320 ..
##  $ RSAL          : Factor w/ 52 levels "Sal1","Sal10",..: 1 1 1 1 1 12 12..
##  $ RGALLER       : Factor w/ 96 levels "Eangus","Eappen",..: 63 63 63 63 ..
##  $ RPAR          : chr  "Pdolic" "Chalci" "Svesic" "Ccruxx" ...
##  $ N_GALLS       : num  22 22 22 22 22 32 32 66 66 66 ...
##  $ NB_GALLS_PAR  : num  9 3 2 5 1 7 8 2 27 1 ...
```

This file provides information of the interaction among gallers, willows and parasitoids. Here, `RSAL`, `RGALLER` and `RPAR` are foreign keys allowing us to retrieve information from files `df_salix`, `df_galler` and `df_parasit`, respectively.

```
df_salix_galler <- readRDS("./rdata/df_salix_galler.rds")
str(df_salix_galler, strict.width="cut")
```

```
## 'data.frame':    2029 obs. of  10 variables:
##  $ RSAL                 : Factor w/ 52 levels "Sal1","Sal10",..: 1 12 23 ..
##  $ RGALLER              : Factor w/ 96 levels "Eangus","Eappen",..: 63 60..
##  $ REARING_NUMBER       : chr  "198714E1-J10elaea" "198714L1-V10bridg" ""..
##  $ LEG                  : chr  "20.07.87" "20.07.87" "22.07.87" "20.07.8"..
##  $ N_GALLS              : num  22 32 66 66 91 92 54 125 129 14 ...
##  $ N_CLEAN GALLS        : num  0 7 0 14 0 8 8 0 12 5 ...
##  $ N_REAL GALLS         : num  22 25 66 52 91 84 46 125 117 9 ...
##  $ PARASITISED_GALLS    : num  20 15 30 34 78 27 24 8 1 1 ...
##  $ UNPARASITISED_GALLS  : num  2 10 36 18 13 57 22 117 116 8 ...
##  $ TOTAL_PARASITISM RATE: num  0.909 0.6 0.455 0.654 0.857 ...
```

This file provides supplementary information on interactions among *Salix* species and sawflies species.

4 Manipulation of files containing nodes and links files

4.1 Binding files together

```
  df_site <- readRDS("./rdata/df_site.rds")
  df_interact <- readRDS("./rdata/df_interact.rds")
  site_interact <- merge(df_site, df_interact, by="REARING_NUMBER")
  head(site_interact)
```

```
##      REARING_NUMBER YEAR_OF_COLL        LEG COUNTRY REGION
## 1 198203W1-R30vimin         1982 1982-06-15 Germany Hessen
## 2 198203W1-R30vimin         1982 1982-06-15 Germany Hessen
## 3 198203W1-R30vimin         1982 1982-06-15 Germany Hessen
## 4 198203W1-R30vimin         1982 1982-06-15 Germany Hessen
## 5  198204A-V10vimin         1982 1982-06-15 Germany Hessen
## 6  198204A-V10vimin         1982 1982-06-15 Germany Hessen
##                      SITE  NDECDEG  EDECDEG ELEVATION  RSAL RGALLER   RPAR
## 1    Kühkopf, Mordhecke I 49.81667 8.416667        87 Sal10  Ovimin Eacicu
## 2    Kühkopf, Mordhecke I 49.81667 8.416667        87 Sal10  Ovimin Etorym
## 3    Kühkopf, Mordhecke I 49.81667 8.416667        87 Sal10  Ovimin Ilappo
## 4    Kühkopf, Mordhecke I 49.81667 8.416667        87 Sal10  Ovimin Ccruxx
## 5 Griesheim bei Darmstadt 49.85000 8.516667        92 Sal10  Ovimin Eacicu
## 6 Griesheim bei Darmstadt 49.85000 8.516667        92 Sal10  Ovimin Etorym
##   N_GALLS NB_GALLS_PAR
## 1      48            5
## 2      48            8
## 3      48            1
## 4      48            5
## 5      51            6
## 6      51            1
```

To include taxonomic information on, for instance, sawflies, we have to merge `df_galler` with `site_interact` using the shared key RGALLER.

```r
df_galler <- readRDS("./rdata/df_galler.rds")
site_interact_wth_taxo <- merge(site_interact, df_galler, by="RGALLER")
head(site_interact_wth_taxo)
```

```
##   RGALLER REARING_NUMBER YEAR_OF_COLL        LEG COUNTRY   REGION
## 1  Eangus  199436XEangus         1994 1994-10-19 Germany   Hessen
## 2  Eappen 1998SZ18Eappen         1998 1998-08-22 Austria Salzburg
## 3  Eappen 1998SZ14Eappen         1998 1998-08-22 Austria Salzburg
## 4  Eappen 1998SZ11Eappen         1998 1998-08-22 Austria Salzburg
## 5  Eappen 1998SZ14Eappen         1998 1998-08-22 Austria Salzburg
## 6  Eappen 1998SZ11Eappen         1998 1998-08-22 Austria Salzburg
##                     SITE  NDECDEG   EDECDEG ELEVATION  RSAL   RPAR N_GALLS
## 1 Kühkopf, Mordhecke III 49.85000  8.383333        85 Sal31   none       1
## 2         Obertauern III 47.23333 13.566667      1656  Sal2 Chalci      10
## 3       Tauernpaß, Tweng 47.18333 13.583333      1266  Sal2 Enobbe      92
## 4         Obertauern III 47.23333 13.566667      1656  Sal2 Pdolic     160
## 5       Tauernpaß, Tweng 47.18333 13.583333      1266  Sal2 Chalci      92
## 6         Obertauern III 47.23333 13.566667      1656  Sal2 Chalci     160
##   NB_GALLS_PAR GENUS                   SPECIES
## 1            0 Euura                   angusta
## 2            5 Euura E. sp. / S. appendiculata
## 3            1 Euura E. sp. / S. appendiculata
## 4            1 Euura E. sp. / S. appendiculata
## 5           50 Euura E. sp. / S. appendiculata
## 6           15 Euura E. sp. / S. appendiculata
##                     GENUS_SPECIES        AUTHOR CODE_GALLTYPE
## 1                   Euura angusta (Hartig 1837)         Sproß
## 2 Euura E. sp. / S. appendiculata    nicht vorh        Knospe
## 3 Euura E. sp. / S. appendiculata    nicht vorh        Knospe
## 4 Euura E. sp. / S. appendiculata    nicht vorh        Knospe
## 5 Euura E. sp. / S. appendiculata    nicht vorh        Knospe
## 6 Euura E. sp. / S. appendiculata    nicht vorh        Knospe
##   CODE_GALLTYPE SIMPLIFIED
## 1               Shoot gall
## 2                 Bud gall
## 3                 Bud gall
## 4                 Bud gall
## 5                 Bud gall
## 6                 Bud gall
```

4.2 B. Turning into matrices

The package `reshape2` allows us to turn long data formats into simple matrices.

```
if(!require(reshape2)){install.packages(reshape2);library(reshape2)}
```

4.2.1 Community matrix

To build a community matrix detailing which species of willows are present in a specific year at a specific site, we write

```
commat_willows <- dcast(SITE+YEAR_OF_COLL~RSAL,data=site_interact,fun.aggregate=length,
  value.var="RSAL")
head(commat_willows[,1:7])
```

```
##                     SITE YEAR_OF_COLL Sal1 Sal10 Sal11 Sal12 Sal13
## 1      Ahrenshoop, Darß         2009    0     0     0     0     0
## 2 Albulapaß nr. Paßhöhe         1999    0     0     0     0     0
## 3 Albulapaß nr. Paßhöhe         2000    0     0     0     0     0
## 4               Aldino         2009    0     5     0     0     0
## 5                 Alta         2001    0     0     0     0     0
## 6      Alta, Baeskades         1988    0     0     0     0     0
```

To achieve a similar tabulation of sawflies present at a specific site in a specific year, we use

```
commat_gallers <- dcast(SITE+YEAR_OF_COLL~RGALLER,data=site_interact,fun.aggregate=length,
  value.var="RGALLER")
head(commat_gallers[,1:7])
```

```
##                     SITE YEAR_OF_COLL Eangus Eappen Eatraa Eaurit Eboreo
## 1      Ahrenshoop, Darß         2009      0      0      0      0      0
## 2 Albulapaß nr. Paßhöhe         1999      0      0      0      0      0
## 3 Albulapaß nr. Paßhöhe         2000      0      0      0      0      0
## 4               Aldino         2009      0      0      0      0      0
## 5                 Alta         2001      0      0      0      0      0
## 6      Alta, Baeskades         1988      0      0      0      0      0
```

To generate an interaction matrix among willows and sawflies, we use

```
df_interact <- readRDS("./rdata/df_interact.rds")
sal_vs_gall <- dcast(RSAL~RGALLER,data=df_interact,fun.aggregate=sum,
  value.var="N_GALLS")
head(sal_vs_gall[,1:7])
```

```
##    RSAL Eangus Eappen Eatraa Eaurit Eboreo Eciner
## 1  Sal1      0      0      0      0      0      0
## 2 Sal10      0      0      0      0      0      0
## 3 Sal11      0      0    695      0      0      0
## 4 Sal12      0      0    872      0      0      0
## 5 Sal13      0      0      0      0      0      0
## 6 Sal14      0      0      0      0      0      0
```

To generate an interaction matrix among sawflies and parasitoids, we write

```
df_interact <- readRDS("./rdata/df_interact.rds")
gall_vs_par <- dcast(RGALLER~RPAR,data=df_interact,fun.aggregate=sum,
  value.var="NB_GALLS_PAR")
head(gall_vs_par[,1:7])
```

```
##   RGALLER Aacumi Aalvea Acubic Adelog Agalli Aminim
## 1  Eangus      0      0      0      0      0      0
## 2  Eappen      1      0      0      0      0      0
## 3  Eatraa     10      0      0      0      0      0
## 4  Eaurit     28      0      0      0      0      0
## 5  Eboreo      0      0      0      0      0      0
## 6  Eciner     22      0      0      0      0      0
```

4.3 Mapping sites

Here we show how to derive the map of the sampling, as shown in Fig. 1A (above).

We first load the packages needed and import the data set:

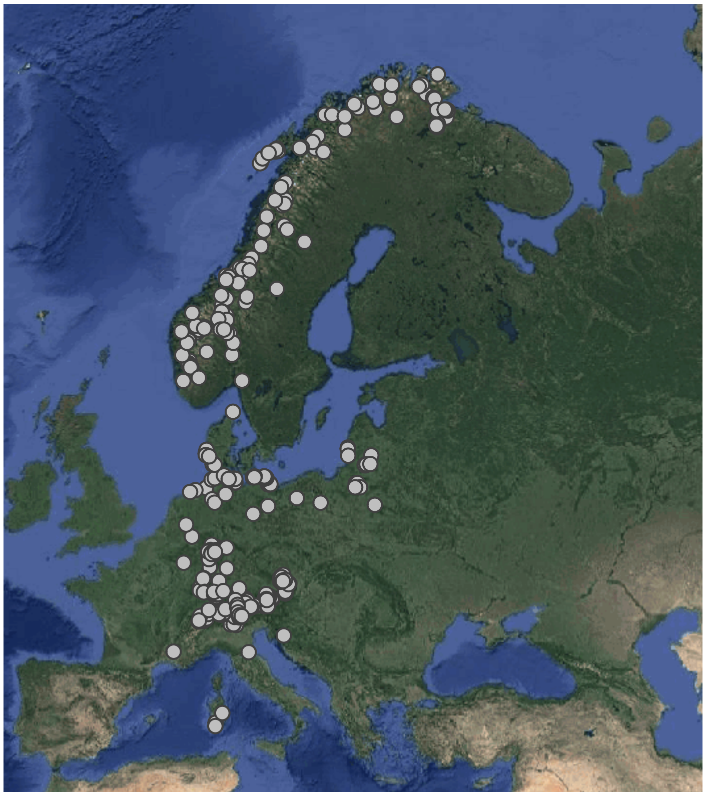
```
library(rgdal)
library(raster)
library(dismo)
```

We then convert the sites into a spatial object (as described in the sp package):

```
df_site <- readRDS("rdata/df_site.rds")[,c(
  "SITE", "NDECDEG", "EDECDEG")] %>% unique
#
sp_site <- SpatialPointsDataFrame(
    df_site[,c("EDECDEG", "NDECDEG")],
    df_site[c("SITE")],
    proj4string = CRS("+proj=longlat +datum=WGS84 +no_defs
     +ellps=WGS84 +towgs84=0,0,0")
    )
```

We adopt a background map available on-line and add our points on top of it:

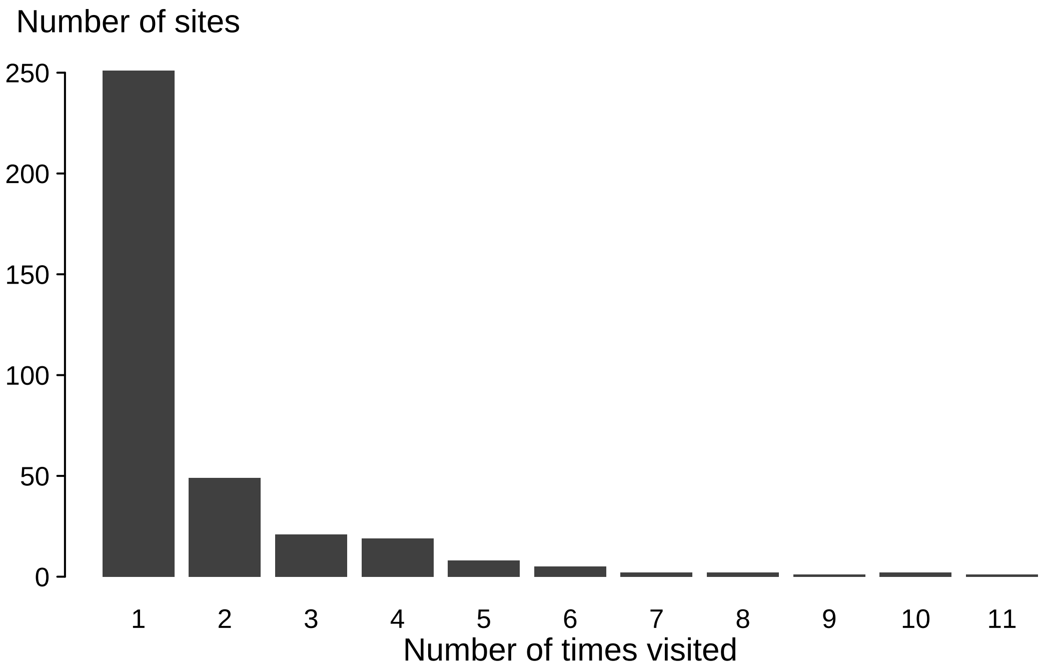
```
## background map
bg_map <- gmap('Europe', type="satellite",
  zoom=3, exp=1.1, scale=2, add=T)
## map
par(mar=c(1,1,1,1))
plot(c(-258698, 4351808), c(4350881, 11745460), asp=1, ann=F, axes=F, type="n")
plot(bg_map, add=TRUE)
plot(spTransform(sp_site, CRS("+init=epsg:3857")), add=TRUE,
  col="grey25", bg="grey75", cex=1, pch=21)
```



4.4 Number of times the sites are visited

We are now able to easily retrieve the number of times a site is visited. Below we do do and display the barplot associated.

```
tmp <- readRDS("rdata/df_site.rds")
par(las=1, cex.axis=.8, mgp=c(2, .4, 0), tcl=-0.2)
cool <- tmp %>% `[`(,c("EDECDEG", "NDECDEG", "YEAR_OF_COLL")) %>%
  unique %>% `[`(,c("EDECDEG", "NDECDEG")) %>%
  apply(1,paste, collapse="/") %>% table %>% table %>%
  graphics::barplot(border=NA, col="grey25")
mtext(1, text= "Number of times visited", line=1.25)
mtext(3, at=-1,  text= "Number of sites", line=.8, adj=0)
```



4.5 Extracting environmental data from `WorldClim`

Converting sites into R spatial object (sp package) offers the possibility to gather environmental data from WorldClim (http://www.worldclim.org/) using the `raster` package. For instance, to retrieve the bioclimatic variables at each site:

We first download the bioclimatic rasters and then extract the values at each site location (using the `sp_site` spatial object previously created).

```
climate <- getData('worldclim', var='bio', res=2.5)
clim_site <- extract(climate,sp_site,df=TRUE)
```

As result, we obtain a dataframe wherein each column corresponds to a bioclimatic variables (http://worldclim.org/bioclim) and each row is a specific site:

```
clim_site <- data.frame(SITE=sp_site@data$SITE,clim_site[,-1])
str(clim_site)
```

```
## 'data.frame':    374 obs. of  20 variables:
##  $ SITE : Factor w/ 374 levels "Ahrenshoop, Darß",..: 60 369 265 370 222 107 103 81 356 266 ...
##  $ bio1 : num  28 0 29 27 -19 80 83 79 81 84 ...
##  $ bio2 : num  81 68 83 84 55 66 64 57 55 68 ...
##  $ bio3 : num  31 29 31 31 27 29 27 25 24 28 ...
##  $ bio4 : num  6173 5686 6251 6233 5252 ...
##  $ bio5 : num  161 119 163 164 87 205 205 187 189 212 ...
##  $ bio6 : num  -94 -108 -97 -99 -115 -22 -26 -40 -33 -27 ...
##  $ bio7 : num  255 227 260 263 202 227 231 227 222 239 ...
##  $ bio8 : num  105 71 106 105 47 151 159 92 96 162 ...
##  $ bio9 : num  -52 -71 -53 -54 -82 31 31 27 28 32 ...
##  $ bio10: num  105 71 106 105 47 155 159 153 155 162 ...
##  $ bio11: num  -52 -71 -53 -54 -82 6 7 6 8 6 ...
##  $ bio12: num  1109 1217 1122 1044 1349 ...
##  $ bio13: num  152 154 153 146 154 81 72 98 98 71 ...
##  $ bio14: num  59 69 60 53 84 45 41 41 41 40 ...
##  $ bio15: num  33 27 33 35 20 19 18 30 30 17 ...
##  $ bio16: num  421 430 425 403 439 225 200 288 288 199 ...
##  $ bio17: num  190 220 195 167 270 138 125 132 132 126 ...
##  $ bio18: num  421 430 425 403 408 222 200 211 211 199 ...
##  $ bio19: num  190 220 195 167 270 177 154 155 155 153 ...
```

4.6 Interaction networks

We next show how to prepare the data to be used in R packages dedicated to network analyses. For this, we first load additional packages:

```r
library(igraph)
library(bipartite)
df_interact <- readRDS("rdata/df_interact.Rds")
```

4.6.1 Using the 'bipartite' package

The lines below creates a contingency table suited for the 'bipartite' package.

```r
bip_salgal <- df_interact[,c("RSAL","RGALLER")] %>% table
bip_galpar <- df_interact[,c("RGALLER","RPAR")] %>% table
```

As an example of the information to be extracted, we compute the C-score using the same package:

```r
C.score(bip_galpar)
```

```
## [1] 0.7891025
```

4.6.2 Using 'igraph'

We here create the metaweb, *i.e* the network including all the interactions described in the dataset. First, we create two networks, one for associations between plants and herbivores, and one for associations between herbivores and their parasitoids, respectively:

```r
mweb_salgal <- df_interact[,c("RSAL","RGALLER")] %>% unique
igr_salgal <- data.frame(
    from = mweb_salgal$RSAL,
    to = mweb_salgal$RGAL
  ) %>% graph_from_data_frame(directed=TRUE)
#
id <- df_interact$RPAR!="none"
mweb_galpar <- df_interact[id,c("RPAR","RGALLER")] %>% unique
igr_salpar <- data.frame(
    from = mweb_galpar$RGAL,
    to = mweb_galpar$RPAR
  ) %>% graph_from_data_frame(directed=TRUE)
```

Then we combine the two networks:

```r
metweb <- igraph::union(igr_salgal, igr_salpar)
```

As an example of the information contained by these matrices, we compute the degree for (*i.e.* number of species associated with) each *Salix* species.

```
igraph::degree(metweb)[1:20]
```

```
##  Sal1  Sal2  Sal3  Sal4  Sal5  Sal6  Sal7  Sal8  Sal9 Sal10 Sal11 Sal12
##     3     4     4     4     3     3     4     3     7     6     4     4
## Sal13 Sal14 Sal15 Sal16 Sal17 Sal18 Sal19 Sal20
##     2     8     3     6     5     8     4     4
```

5 References

1. R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.