A real life workflow for the TR8 package

Gionata Bocci
Pisa (ITALY)
boccigionata@gmail.com

December 1, 2020

1 A 'real life' workflow

In this vignette I will describe a typical workflow for a researcher interested in using the `TR8` package: this is meant to be a step-by-step guide involving most of the common problems that are faced in importing data, checking them and running `tr8`[1]. This section relies on the dataset used by Sandau et al.(2014)[4] which is publicly available at `dryad`[3] (under a CC0 1.0 licence). **NOTA BENE**: this dataset is here merely used as a tool to demonstrate some important points related to the functioning of `TR8`, thus what is proposed in this tutorial should be considered just as a *proof of concept* (i.e. I am not suggesting that the species' names in the original dataset should be changed neither I am proposing a different statistical analysis for the data).

1.1 Retrieve original data

The dataset contains `total above-ground biomass` data of species sampled in 12 wildflower strips in Switzerland. The experimental factors are:

SDiv Species richness of sown mixtures (2,6,12,20)

Treat Control of herbivores and predators (C: control, PE: exclusion of predators, PHE: exclusion of predators and herbivores)

The dataset is available as a `xlsx` file (which is a common case); several alternatives are available to import this kind of files into `R` (e.g. you can download the dataset and load it into an `R` session or you could save a `.csv` version of the file and then load it into `R` using `read.csv()`; if you are following these strategies, feel free to skip this paragraph); for this tutorial we will use `readxl`[5] to download and load the dataset.

[1] The process described here is rather lengthy in order to describe each single step in detail; users who are confident in the use of R could make most of the steps much shorter than what's presented here.

```
> ## the readxl package is needed
> ## library(readxl)
> ## store the  url of the dryad package
> url<-"http://datadryad.org/bitstream/handle/
+     10255/dryad.65646/MEE-13-11-651R2_data.xlsx?sequence=1"
> ## choose the extension for the temp file where
> ## data will be stored
> tmp = tempfile(fileext = ".xlsx")
> ## download the data
> download.file(url = url, destfile = tmp)
> ## we first read the "metadata" sheet from the xlsx file
> ## (the row containing the species names start from
> ## row 13
> metadata<-read_excel(path=tmp,sheet="metadata",skip=12,col_names=F)
> ## lets rename the column of this dataset
> names(metadata)<-c("Col1","Col2")
> ## then read the vegetation data
> veg_data <-readWorksheetFromFile(file = tmp, sheet = "data.txt")
> ## only the columns from 11 to 123 contains the species data
> veg_data<-veg_data[,11:123]
>        ## round veg_data numbers to the second digit
> veg_data<-round(veg_data,digits = 2)
> ## read the dataset with the environmental variables
> env_data<-read_excel(path = tmp, sheet = "data.txt")
> ## and select only the column from 1 to 4 which contain
> ## the data of interest
> env_data<-env_data[,1:4]
```

We now have the following dataframes:

metadata contains two columns, `Col1` contains short codes used by the authors as surrogates of the full scientific names of species for the species which are stored in the `Col2` column

veg_data contains a $sites * species$ table of biomass values

env_data contains the experimental variables

1.2 Check species names

The first suggested step is to check species names using the `taxize` package in order to see whether there are misspelled names; the `tnrs` function accepts a vector of plant species names and tries to match them with *accepted* scientific names; the function returns a dataframe with various columns: in the column `score` each entry is given a score according to the level of "resemblance" with correct names; the score is "1" if the name is correct, less than "1" if some

problems with the name are found[2]. **NOTA BENE**: from the `tnrs` help page "If there is no match in the Taxosaurus database, nothing is returned, so you will not get anything back for non matches" thus we should worry of both "less than 1" scores **AND** missing entries in the dataframe returned by `tnrs`.

```
> library(taxize)
> check_names<-tnrs(metadata$Col2,source="iPlant_TNRS")
```

Check now if there are species which were discarded by `tnrs` output since they were not found in reference databases.

```
> setdiff(metadata$Col2,check_names$submittedname)
```

The results is 0, thus `tnrs` found at least a partial match for all the species names we provided. Next we should check which species got a `score` which is less than 1.

```
> issues<-with(check_names,check_names[score!="1",])
> issues[,c("submittedname","score","acceptedname","authority")]
```

submittedname	score	acceptedname	authority
Poaceae (undetermined)	0.9	Poaceae	Barnhart
Epilobium sp.	0.9	Epilobium	L.
Cerastium sp.	0.9	Cerastium	L.
Fallopia convolvulus (L.) A. Löwe	0.96	Fallopia convolvulus	(L.) Á. Löve
Festuca sp.	0.9	Festuca	L.
Chenopodium sp.	0.9	Chenopodium	L.
Phleum pratense agg.	0.9	Phleum pratense	L.
Polygonum sp.	0.9	Polygonum	L.
Polygonum mite (=Persicaria laxiflora)	0.9	Persicaria mitis	(Schrank) Assenov
Rubus sp.	0.9	Rubus	L.
Juncus sp.	0.9	Juncus	L.
Orobanche sp.	0.9	Orobanche	L.
Triticum sp.	0.9	Triticum	L.
Taraxacum officinale!!!!!	0.9	Taraxacum	F.H. Wigg.
Setaria pumila (Poir.) Schult.	0.96	Setaria pumila (Poir.) Roem. & Schult.	

We observe here some important points:

- for some species only the Genus is present, thus `tr8` function will not be able to return traits values for those species; those species names can't be used with `tr8`;
- in the case of *Taraxacum officinale!!!!!* we should remove the extra `"!"` from the species names;
- *Polygonum mite* is not recognized as an accepted name; for this tutorial we assume that it should be changed to *Persicaria mitis*;

[2]Note that `score` contains strings, not numbers, thus we cannot use the "score<1" condition, but must rely on searching those strings which are "different from 1"

- the aggregate species `Phleum pratense agg.` poses some problems: for this tutorial we decide to accept it as *Phleum pratense* L.;
- for *Fallopia convolvulus* and *Setaria pumila* the issues are related to authors names (for *F. convolvulus* "A." should be accented ("Á."), while in *S. pumila*, `Roem.` author name is missing).

The last point is not strictly relevant for using the `tr8` function since authors' names should not be included in the list of species names passed to the function (but we correct the authors names anyway so that when we re-run `tnrs` function, no mistakes are found for these entries).

We adopt the following fixes for the issues we've found:

```
> library(plyr)
> ## we use the revalue function in the plyr package
> ## to fix all the above mentioned issues
> metadata$Col2<-revalue(metadata$Col2,
+      c("Taraxacum officinale!!!!!"="Taraxacum officinale F.H. Wigg."))
> metadata$Col2<-revalue(metadata$Col2,
+      c("Polygonum mite (=Persicaria laxiflora)"="Persicaria mitis (Schrank) Assenov"))
> metadata$Col2<-revalue(metadata$Col2,
+      c("Fallopia convolvulus (L.) A. Löwe"="Fallopia convolvulus (L.) Á. Löve"))
> metadata$Col2<-revalue(metadata$Col2,
+      c("Setaria pumila (Poir.) Schult."="Setaria pumila (Poir.) Roem. & Schult."))
> metadata$Col2<-revalue(metadata$Col2,
+      c("Phleum pratense agg."="Phleum pratense L."))
```

And re-run the `tnrs` function as a cross-check:

```
> check_names<-tnrs(metadata$Col2,source="iPlant_TNRS")
> issues<-with(check_names,check_names[score!="1",])
> issues[,c("submittedname","acceptedname","score")]
```

```
         submittedname acceptedname score
Poaceae (undetermined)      Poaceae   0.9
         Epilobium sp.    Epilobium   0.9
         Cerastium sp.    Cerastium   0.9
           Festuca sp.      Festuca   0.9
       Chenopodium sp.  Chenopodium   0.9
         Polygonum sp.    Polygonum   0.9
             Rubus sp.        Rubus   0.9
            Juncus sp.       Juncus   0.9
         Orobanche sp.    Orobanche   0.9
          Triticum sp.     Triticum   0.9
```

We observe now that only those entries which were identified at the Genus level raise some issues; my suggestion is to remove them (i.e. those entries for which the score is "0.9") from the original `metadata` dataframe.

First we merge the original *metadata* dataframe with *check_names* so that we have original names (with short codes used by authors) and the corrected ones in a single object:

```
> final_dataframe<-merge(metadata,check_names,
+          by.x = "Col2",by.y="submittedname")
```

Then we exlude those species which were identified at the Genus level, i.e. those contained in the *issues* dataframe:

```
> final_dataframe<-final_dataframe[
+         !final_dataframe$Col2%in%issues$submittedname,]
```

In this way we now have the `final_dataframe` data frame in which each entry has both the name present in the original dataset and the correct scientific names (`acceptedname`), which should be passed to the `tr8` function.

1.3 Use the `tr8` function

We can now use the `tr8` function; suppose we are interested in downloading the following traits:

- `Maximum height`
- `Leaf area`
- `Leaf mass`
- `Life form`
- `Strategy type`

Observing the `available_tr8` dataframe (simply write `available_tr8` in the R console), we see the following correspondences:

`Maximum height` is available in Ecoflora and its short code is `h_max`,

`Leaf area` is available in Ecoflora and its short code is `le_area`,

`Leaf mass` is available in LEDA and its short code is `leaf_mass`,

`Life form` is available in BiolFlor[3] and its short code is `li_form_B`,

`Strategy type` is available in BiolFlor and its short code is `strategy`

Thus the `tr8` should be run as follows (**beware**: this may take some time!)

```
> species_names<-final_dataframe$acceptedname
> my_traits<-c("h_max","le_area","leaf_mass","li_form_B","strategy")
> retrieved_traits<-tr8(species_list = species_names,download_list = my_traits)
```

[3]Life form is also avaialbe in Ecoflora; here we opt for BiolFlor in order to increase the range of queried databases

The results are reported in table 1: the table here shown is a condensed version of what you will get with the above commands: the traits returned from BiolFlor contain strings within brackets which explain the value of each trait (e.g. "csr (competitors/stress-tolerators/ruderals)"); these are useful for interpreting the data, but made our table clumsy, thus I removed them).
Having a look at the table, some points should be noted:

A) `h_max` for *C. arvensis* reports two values (75;10) since because Ecoflora provides two values for this species; it is left to the user to decide which one should be used! (you may decide to calculate a mean)

B) for some species `le_area` reports more than one class (e.g. for *A. pseudoplatanus* we have 0-100;100-1000): here again the user should decide how to cope with this ambiguity.

In order to be able to carry on some statistical analysis, we adopt the following decisions:

- for *C. arvensis* we calculate the mean between 75 and 10 and use that as the `h_max`

```
> ## we extract the data from the object returned by tr8()
> traits<-extract_traits(retrieved_traits)
> ## first I convert the column to character
> traits$h_max<-as.character(traits$h_max)
> traits$h_max[which(row.names(traits)=="Convolvulus arvensis")]<-"42.5"
```

- the `h_max` column can now be converted to a `numeric` variable (it should be treated as such in a statistical analysis)

```
> traits$h_max<-as.numeric(traits$h_max)
```

- for `le_area` we propose to use the midpoint of the range indicated for each species (e.g. *A. stolonifera* has a `le_area` range of 1-10, thus it will be assigned a value of 5.5); for those species who have 2 ranges, the common value among those ranges will be the selected one (e.g. *R. obtusifolius* has two ranges, 100-1000 and 10-100, thus 100 will be the selected value.

```
> traits$le_area<-revalue(traits$le_area,
+       c("0.1-1"=0.55,
+         "1-10"=5.5,
+         "10-100"=55,
+         "100-1000"=550,
+         "1-10;0.1-1"=1,
+         "10-100;1-10"=10,
+         "100-1000;10-100"=100,
+         "10-100;100-1000"=100))
> ## and convert them to numeric
> traits$le_area<-as.numeric(as.character(traits$le_area))
```

- We recode `li_form_B` values in order to have shorter labels on the graphs:
  - "C (Chamaephyte) - H (Hemicryptophyte)" becomes "C - H"
  - "G (Geophyte)" becomes "G"
  - "G (Geophyte) - H (Hemicryptophyte)" becomes "G - H"
  - "H (Hemicryptophyte)" becomes "H"
  - "H (Hemicryptophyte) - T (Therophyte)" becomes "H - T"
  - "M (Macrophanerophyte)" becomes "M"
  - "M (Macrophanerophyte) - N (Nanophanerophyte)" becomes "M - N"
  - "T (Therophyte)" becomes "T"

```
> traits$li_form_B<-revalue(traits$li_form_B,
+   c("C (Chamaephyte) - H (Hemicryptophyte)"="C - H",
+     "G (Geophyte)"="G",
+     "G (Geophyte) - H (Hemicryptophyte)"="G - H",
+     "H (Hemicryptophyte)"="H",
+     "H (Hemicryptophyte) - T (Therophyte)"="H - T",
+     "M (Macrophanerophyte)"="M",
+     "M (Macrophanerophyte) - N (Nanophanerophyte)"="M - N",
+     "T (Therophyte)"="T"))
> ## convert it to factor
> traits$li_form_B<-as.factor(traits$li_form_B)
```

- We also recode `strategy` values:
  - "c (competitors)" becomes "c"
  - "cr (competitors/ruderals)" becomes "cr"
  - "cs (competitors/stress-tolerators)" becomes "cs"
  - "csr (competitors/stress-tolerators/ruderals)" becomes "csr"
  - "r (ruderals)" becomes "r"

```
> traits$strategy<-revalue(traits$strategy,c("c (competitors)"="c",
+  "cr (competitors/ruderals)"="cr",
+  "cs (competitors/stress-tolerators)"="cs",
+  "csr (competitors/stress-tolerators/ruderals)"="csr",
+  "r (ruderals)"="r"))
> traits$strategy<-as.factor(traits$strategy)
```

2 An example of analysis

Plant traits data are often used to assess species or community-level trait response to environmental gradient. Several statistical techniques can be applied: Kleyer et al. [2] present an interesting classification of various techniques and propose some criteria to choose the best technique according to the research aims.
In this paragraph I present an example of a series of statistical analyses applied to trait data. The analyses presented here follow the very same steps described by Kleyer et al. [2] in the Supporting information to their paper, thus readers are strongly encouraged to read that publication. In this case we will adopt the **RLQ** analysis, thus the following data are required:

R the "environment" dataset: this corresponds to `env_data` dataframe (please note: for the sake of having a complete example of analysis, we are here assuming that the two variables included in the dataset represent *environmental variables*)

L is the *species*sites* dataset (our `veg_data` dataframe)

Q is the *trait*species* dataframe (our `trait` dataframe)

In order be able to run the desired analyses, our datasets need a few more fixes.
Please note that I will follow a trial&error approach here, showing what is likely to go wrong and what solutions should be adopted.

1. We change species names in the `traits` dataframe switching back to the short codes so that the same names are used in `traits` and `veg_data`:

```
> row.names(traits)<-mapvalues(row.names(traits),
+    from=final_dataframe$acceptedname,to=final_dataframe$Col1)
```

2. In the `traits` dataframe, we were not able to find trait data for some species: this will cause problems for the algorithm we will use, thus we need to remove those species from the datasets (this will impair the results of the whole analysis, but for the moment we do not have other choices):

```
> traits<-traits[complete.cases(traits),]
```

3. **Beware:** the same species should be present in **L** and **Q** tables, thus we should remove from `veg_data` those species which were eliminated from `traits` with the previous operation:

```
> vegetation<-veg_data[,names(veg_data)%in%row.names(traits)]
```

4. we can now (try to) perform the first multivariate analyses; first we use a Correspondance analysis on the **L** dataset (the `vegetation` dataframe); for this we will need the `ade4` package[1]:

```
> library(ade4)
> coa<-dudi.coa(vegetation,scannf=F)
```

5. Then we need to perform and ordination on our `trait` dataframe; Kleyer et al.[2] used a Principal component analysis (`dudi.pca` in `ade4`); in our case the `trait` dataframe contains both quantitative and factors variables, thus the `dudi.pca` can't be used; we therefore opt for `dudi.hillsmith` which accepts both type of variables.

```
> hil.traits<-dudi.hillsmith(traits,row.w=coa$cw,scannf = FALSE)
```

which returns:

```
Error in eigen(df, symmetric = TRUE) : infinite or missing values in 'x'
```

The problem here is due to the fact that in the `vegetation` dataset there are some species (columns) which have zero values in all the samplings (rows); these species should be removed:

```
> ##select which columns have at least one non-zero value
> selection<-colSums(vegetation)>0
> ## and now we choose only those columns
> vegetation<-vegetation[,selection]
```

6. As stressed before, `L` and `Q` matrix should have the same species, thus we should be sure that those species with all-zero-values are removed from `traits` as well:

```
> traits<-traits[row.names(traits)%in%names(vegetation),]
```

7. Let's also order species names in both dataframes:

```
> vegetation<- vegetation[,order(names(vegetation))]
> traits<-traits[order(row.names(traits)),]
```

8. We now re-perform the `pca` on `vegetation` and `dudi.hillsmith` on `traits` datasets:

```
> coa<-dudi.coa(vegetation,scannf=F)
> traits.hill<-dudi.hillsmith(traits,row.w=coa$cw,scannf = F)
```

9. And perform `dudi.hillsmith` on the `env_data` dataset as well:

   ```
   > env.hill<-dudi.hillsmith(env_data,row.w=coa$lw,scannf = FALSE)
   ```

   which returns:

   ```
   Error in x * w : non-numeric argument to binary operator
   ```

   This is due to the fact that the `Treat` variable in `env_data` is of class "character", while `dudi.hillsmith` accepts either a quantitative or a factor variable; we thus should convert the variable to a factor:

   ```
   > env_data$Treat<-as.factor(env_data$Treat)
   ```

   Now we re-run the analysis:

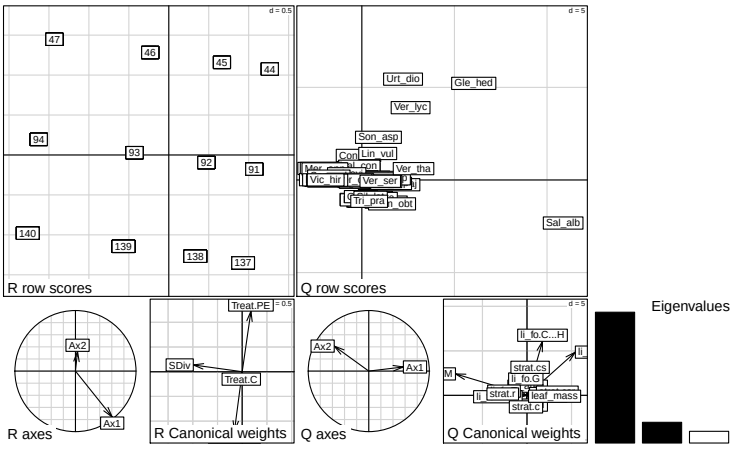
   ```
   > env.hill<-dudi.hillsmith(env_data,row.w=coa$lw,scannf = FALSE)
   ```

10. And (finally) we can perform the `rlq` analysis using the previusly created objects:

    ```
    > rlq_tr8<-rlq(env.hill,coa,traits.hill,scannf = F)
    ```

11. The object created can be passed to the `plot` function

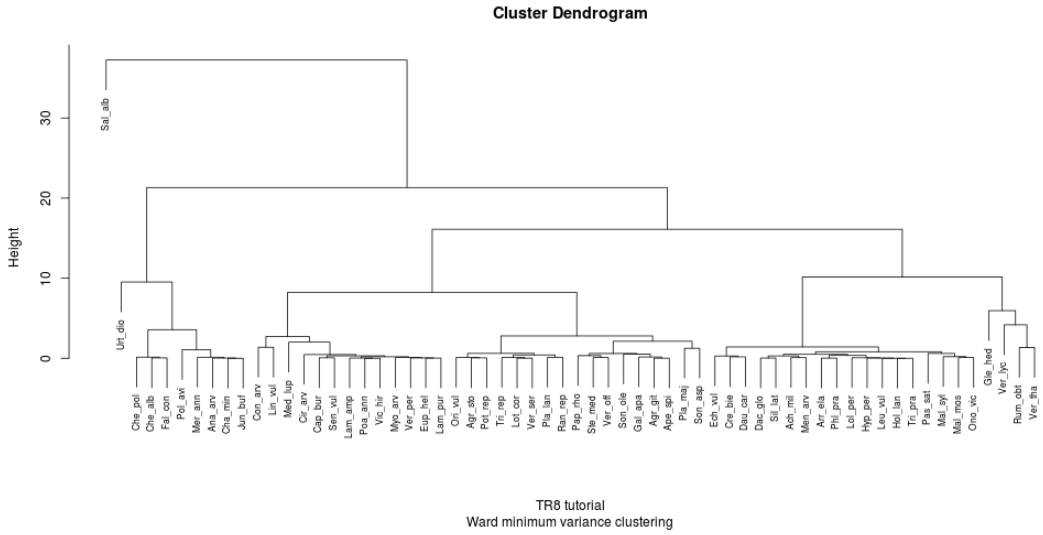
    ```
    > plot(rlq_tr8)
    ```



Several plotting functions are available for `ade4` objects, please refer to the package documentation for further detailed information.

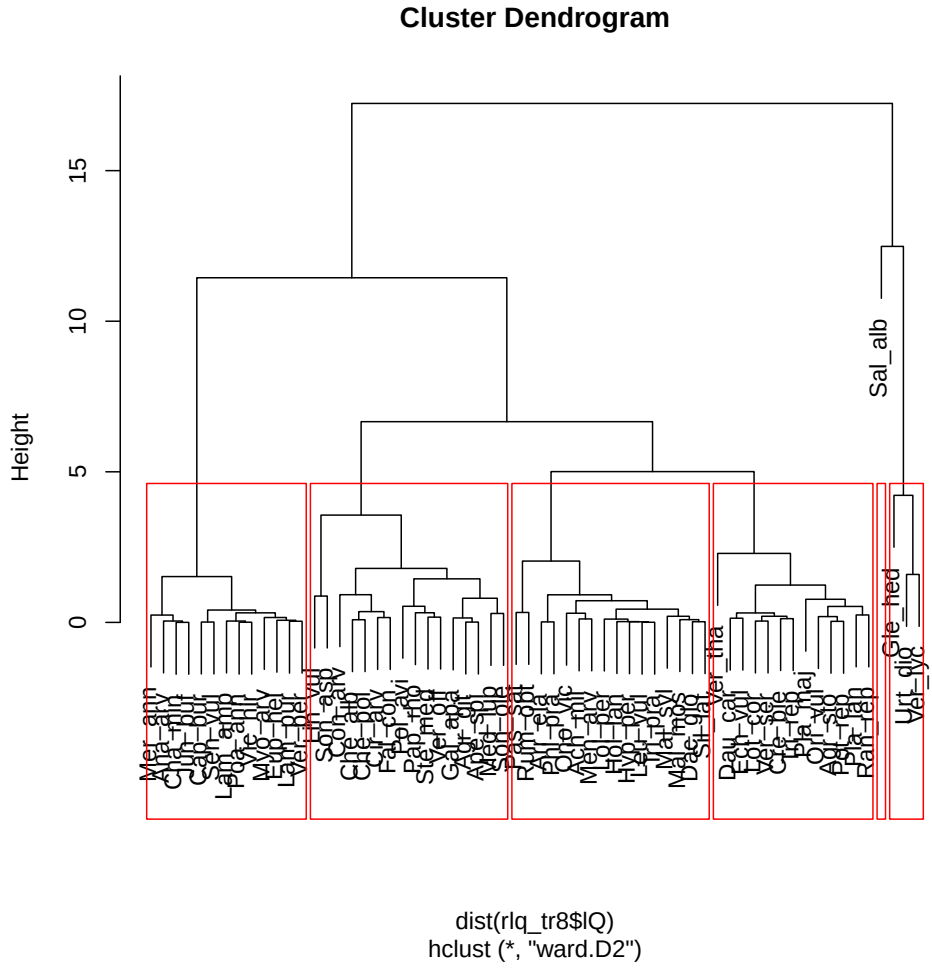
12. We now perform a clustering and its results will be used for building functional groups. Following the example provided by Kleyer et al. [2] we use the Ward minimum variance clustering:

```
> clust<-hclust(dist(rlq_tr8$lQ),method="ward.D2")
> plot(clust,sub="Ward minimum variance clustering",xlab="TR8 tutorial")
```



13. We should now decide where the dendrogram should be cut: several criteria exist: in this case (not to over-complicate the tutorial), based on the visual estimation of the graph, we decide that 6 main groups are present: 2-single species ones (i.e. *S. alba* and *U. dioica*) and 4 multi-species groups. We can thus cut our plot:

```
> rect.hclust(clust,k=6)
```

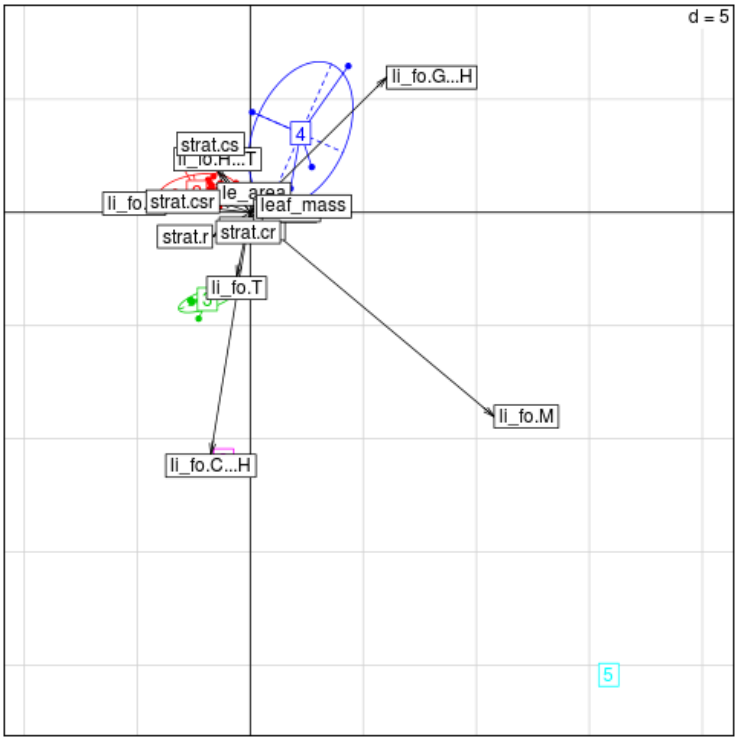


And save the results of this *cutting* for later graphs:

```
> cuts<-cutree(clust,6)
```

14. We can now plot an ordination graph of the species, grouped according to the newly defined functional groups and plot, on top of that graph, traits data:

```
> s.class(rlq_tr8$lQ,as.factor(cuts),col=1:6)
> s.arrow(rlq_tr8$c1,add.plot = TRUE)
```



15. And finally the created grouping can be interpreted in terms of traits (the results can be seen in figure at page 15):

```
> par(mfrow=c(3,2))
> plot(traits$h_max~as.factor(cuts),main="Maxim height",
+      ylab="max height",border = 1:6,xlab="Group number")
> plot(traits$le_area~as.factor(cuts),main="Leaf area",
+      ylab="leaf area",border = 1:6,xlab="Group number")
> plot(traits$leaf_mass~as.factor(cuts),main="Leaf mass",
+      ylab="leaf mass",border = 1:6,xlab="Group number")
> plot(table(cuts,traits$strategy),main="CSR strategy",
+      ylab="strategy",border = 1:6,xlab="Group number")
> plot(table(cuts,traits$li_form_B),main="Life form",
+      ylab="life form",border = 1:6,xlab="Group number")
> par(mfrow=c(1,1))
```

From the graphs we are able to say that *group 1* is composed of only "Hemicryptophyte" species, while *group 2* has a more homogeneous combination of "Geophytes", "Hemicryptophytes" and "Hemicryptophyte - Therophyte" species. If we look at `maximum height`, there's a group which is clearly different from all the rest: this is the one which is composed of only *Salix alba*: this is the only woody species in our analyzed dataset,

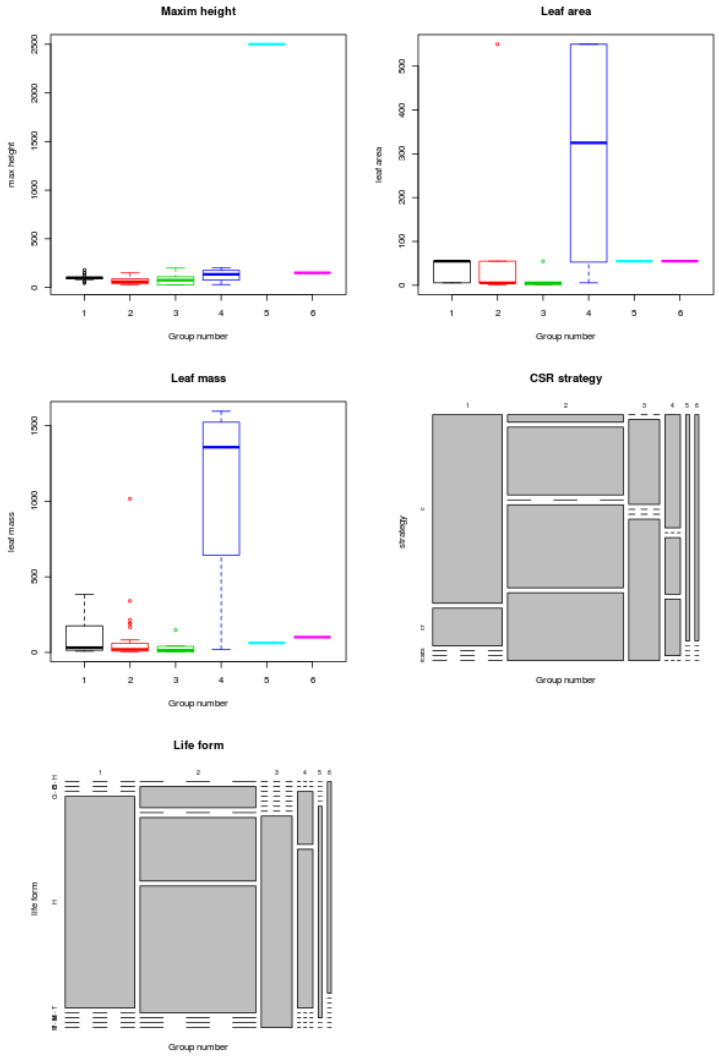


Figure 1: Results of traits ditribution in defined functional groups.

thus it's clear that, when considering the `maximum height` trait, it would be separated from all the other species. In terms of `leaf area`, `group 4` is composed of some of the broadleaf species which have the maximum values for his trait (e.g. *Verbascum thapsus* and *V. lychnitis*).

3 Some concluding remarks

What has been presented above is just an example of a workflow, thus the results obtained and the graphs presented should be considered as potential outcomes of the process, not as valid scientific results.
There are also many points which were not included in the analysis and should be considered in order to make it more "scientifically sound"; among the most important ones, I list the following:

- We did not transform original data: it would have made sense to transform the original abundance data, e.g. using standardization algorithm provided by the `decostand()` function in `vegan`.
- In the end we excluded many species from our analysis since we had not found trait data for them; we could have searched for data in different databases (e.g. `life_form` is also available in `Ecolfora`) or we could have got data from other sources - like books, floras, etc... - and added those values to the downloaded traits).
- We used trait values retrieved from databases[4] and assumed that those were correct for our analysis; this poses some problems, e.g the case of *S. alba* is striking in this respect: looking at the abundance data, it's clear that what were sampled were small individuals, but using data retrieved by `tr8` we used a value of 25 m for its `maximum height` value.
- `Leaf area` values could have been coded as ordered factors, instead of using midpoint values of each class (which is probably one of the reason for the shape of the boxplot in figure at page 15).

We could repeat all the analyses taking these points (and many other relevant ones) into account, but this, as in any tutorial which deserves its name, is left to the reader as an exercise.

References

[1] S. Dray and A.B. Dufour. The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22(4):1–20, 2007.

[2] Michael Kleyer, Stéphane Dray, Francescode Bello, Jan Lepš, Robin J. Pakeman, Barbara Strauss, Wilfried Thuiller, and Sandra Lavorel. Assessing

[4] This is what `TR8` was created for...

species and community functional responses to environmental gradients: which multivariate methods? *Journal of Vegetation Science*, 23(5):805–821, October 2012.

[3] N Sandau, RP Rohr, RE Naisbit, Y Fabian, OT Bruggisser, P Kehrli, A Aebi, and L Bersier. Data from: Including community composition in biodiversity-productivity models, 2014.

[4] Nadine Sandau, Rudolf P. Rohr, Russell E. Naisbit, Yvonne Fabian, Odile T. Bruggisser, Patrik Kehrli, Alexandre Aebi, and Louis-Félix Bersier. Including community composition in biodiversity–productivity models. *Methods in Ecology and Evolution*, 5(8):815–823, 2014.

[5] Hadley Wickham. *readxl: Read Excel Files*, 2015. R package version 0.1.0.

	h_max	le_area	li_form_B	strategy	leaf_mass
Acer pseudoplatanus	3000	10-100;100-1000	M	c	NA
Achillea millefolium	45	1-10	H	c	21.76
Aethusa cynapium	120		H - T	cr	118.21
Agrostemma githago	100	1-10	H - T	cr	57.23
Agrostis stolonifera	100	1-10	H	csr	9.41
Amaranthus retroflexus			T	cr	265.49
Anagallis arvensis	30	0.1-1	T	r	2.76
Apera spica-venti	100	1-10	H - T	cr	43.95
Arrhenatherum elatius	150	10-100	H	c	30.65
Brassica napus	100		T	cr	NA
Campanula patula	60	1-10	H	csr	NA
Capsella bursa-pastoris	40	10-100	H - T	r	60.89
Centaurea cyanus	90		H - T	cr	33.62
Centaurea jacea			H	c	NA
Chaenorhinum minus	25	0.1-1	T	r	3.46
Chenopodium album	100	1-10	T	cr	148.55
Chenopodium polyspermum	100	1-10	T	cr	36.84
Cichorium intybus	120		H	c	556.12
Cirsium arvense	90	10-100	G	c	190.54
Convolvulus arvensis	75;10	1-10	G	cr	40.73
Conyza canadensis	100		H - T	cr	24.74
Cota tinctoria			H	cs	NA
Crepis biennis	120	10-100	H	cr	189.01
Dactylis glomerata	100	10-100	H	c	49.9
Daucus carota	100	10-100	H	cr	12.84
Digitaria sanguinalis			T	r	8.62
Dipsacus fullonum	150		H	cr	1118.24
Echinochloa crus-galli			T	cr	76.9
Echium vulgare	90	10-100	H	cr	282.5
Elytrigia repens subsp. repens	120	10-100	G - H	c	NA
Equisetum arvense	80		G	cr	178.64
Euphorbia helioscopia	50	1-10	H - T	r	12.88
Euphorbia stricta	80	1-10	H - T	r	NA
Fallopia convolvulus	120	10-100	T	cr	44.26
Galinsoga quadriradiata	80	1-10	T	cr	NA
Galium aparine	120	1-10;0.1-1	H - T	cr	5.7
Galium mollugo subsp. album			H	c	NA
Geranium rotundifolium	40	1-10	T	r	NA
Glechoma hederacea	30	1-10	G - H	csr	20.96
Gnaphalium uliginosum	20	1-10;0.1-1	T	r	NA
Holcus lanatus	100	10-100;1-10	H	c	21.8
Hypericum perforatum	100	1-10	H	c	8.72
Juglans regia			M	c	NA
Juncus bufonius	25	0.1-1	T	r	4.55

Kickxia elatine	50	1-10	T	r	NA
Kickxia spuria	50	1-10	T	r	NA
Lactuca serriola	200		H - T	cr	397.12
Lamium amplexicaule	25	1-10	H - T	r	8.46
Lamium purpureum	45	1-10	H - T	r	11.32
Leucanthemum vulgare	100	1-10	H	c	27.12
Linaria vulgaris	80	1-10	G	csr	5.68
Lolium perenne	90	1-10	H	c	14.53
Lotus corniculatus	35	1-10	H	csr	4.5
Malva moschata	80	10-100	H	c	176.06
Malva sylvestris	90	10-100	H	c	341.6
Matricaria chamomilla	60		H - T	r	NA
Medicago lupulina	60	1-10	H - T	csr	17.28
Medicago sativa	90	1-10	NA	NA	29.19
Melilotus albus	150	1-10	H - T	cr	NA
Mentha arvensis	60	1-10	H	c	14.21
Mercurialis annua	50	1-10	T	r	17.07
Myosotis arvensis	60	1-10	H - T	r	28.22
Oenothera biennis	100	10-100	H	cr	NA
Onobrychis viciifolia	60	10-100	H	c	164.05
Origanum vulgare	80	1-10	H	csr	23.37
Oxalis stricta	40	1-10	NA	NA	NA
Papaver rhoeas	60	10-100	H - T	cr	166.81
Pastinaca sativa	180	10-100	H	c	382.98
Persicaria mitis			NA	NA	NA
Phleum pratense	150	10-100	H	c	9.91
Plantago lanceolata	40	10-100	H	csr	82.79
Plantago major	60	10-100	H	csr	1016.91
Poa annua	30	1-10	H - T	r	3.21
Polygonum aviculare	200	1-10	T	r	13.01
Potentilla reptans	100	1-10	H	csr	53.72
Prunella vulgaris	30	1-10	C - H	csr	16.11
Ranunculus repens	60	10-100	H	csr	193.94
Rumex obtusifolius	120	100-1000;10-100	H	c	1449.89
Sagina apetala	10		T	r	0.32
Salix alba	2500	10-100	M	c	63.36
Salix caprea	1000	10-100;1-10	M - N	c	NA
Scrophularia nodosa	80	10-100	H	cs	313.6
Senecio vulgaris	45	10-100	H - T	r	17.59
Setaria pumila			T	r	75.09
Silene latifolia	100	10-100	H	c	78.06
Sinapis alba	80	10-100	T	cr	84.21
Solanum americanum	60		T	r	NA
Sonchus arvensis	150		H	cr	549.53
Sonchus asper	150	100-1000	H - T	cr	216.37
Sonchus oleraceus	150	10-100	H - T	cr	341.14

Stellaria media	40	1-10	H - T	cr	8.75
Tanacetum vulgare	120		H	c	427.99
Taraxacum officinale	30		NA	NA	NA
Trifolium pratense	100	1-10	H	c	32.58
Trifolium repens	50	1-10	H	csr	11.76
Urtica dioica	150	10-100	C - H	c	101.22
Verbascum lychnitis	150	100-1000	H	cs	1265
Verbascum thapsus	200	100-1000	H	c	1595.28
Verbena officinalis	60	1-10	H - T	cr	26.92
Veronica persica	40	1-10	H - T	r	13.71
Veronica serpyllifolia	30	0.1-1	H	csr	8.02
Vicia hirsuta	30	1-10	H - T	r	10.51
Viola arvensis	45		H - T	r	23.1

Table 1: Traits data collected through the `tr8` function; please note that to improve the readability of the table, in `li_form_B` and `strategy` the traits explanation between brackets were removed (e.g. "cr (competitors/ruderals)" was converted to "cr" and "M (Macrophanerophyte) - N (Nanophanerophyte)" to "M - N".