Pipeline for functional assignation of heterogeneous data

Anne-Sophie Masson
annesomasson@proton.me

IFOSSA
project link

General info

Objectives of the pipeline

- to create objects usable for the analysis of ecological groups of organisms from heterogeneous data
  The *gratin* package (Le Guillarme, N., Hedde, M., Potapov, A. M., Martínez-Muñoz, C. A., Berg, M. P., Briones, M. J. I., Calderón-Sanou, I., Degrune, F., Hohberg, K., Martinez-Almoyna, C., Pey, B., Russell, D. J., & Thuiller, W. (2023). The Soil Food Web Ontology : Aligning trophic groups, processes, resources, and dietary traits to support food-web research. Ecological Informatics, 78, 102360. https://doi.org/10.1016/j.ecoinf.2023.102360 + Le Guillarme, N., & Thuiller, W. (2023). A practical approach to constructing a knowledge graph for soil ecological research. European Journal of Soil Biology, 117, 103497. https://doi.org/10.1016/j.ejsobi.2023.103497) and the standardized structure of *phyloseq* objects (https://joey711.github.io/phyloseq/import-data) are used.
- to pre-analyse data (some figures and tables to check outputs)

3 constructors are needed to create *phyloseq* objects (*PO*) :

- ***tidy_data*** (*otu_table*) = the tidy table of abundance
- ***tax_table*** (*tax_table*) = the cleaned taxonomy table
- ***fun_table*** (to be combined with *tax_table* later) = the functional table generated with *gratin* package
- ***sam_data*** = metadata

There are several steps to create these constructors:

- step A = data homogenization

→ ***tidy_data*** creation

- step B = taxonomic cleaning and filtering

→ ***tax_table*** creation

- step C = functional assignation

→ ***fun_table*** creation

- step D = merge into *phyloseq* objects

→ ***sam_data*** and ***PO*** creation

⇒ open the script

Some details of the steps

- step A = data homogenization

community_0 = raw data (from observational counts or metabarcoding abundances)
tidy_community_1 = homogenized format, structure and annotations
tidy_community_2 = with minimal information = ***tidy_data*** by community
tidy_ALL = all communities combined = ***tidy_data*** with all communities

Replace community by :

- macrofauna_surface
- macrofauna_aerial
- macrofauna_foliar
- nematodes
- micro_arthropodes
- microorganisms
  - bacteria
  - fungi
  - protists

Some details of the steps

- step B = taxonomic assignation, cleaning and filtering

For microorganisms, done during step A :
# cleaning step 1 = remove “no data”
# cleaning step 2 = modify taxonomic names if needed
# cleaning step 3 = filter
## Filter 1 (based on taxonomy) = keep Bacteria and remove Chloroplasts, keep Fungi or keep Cercozoa depending on the targeted community and amplicon sequences
## Filter 2 (based on abundance) = remove low abundant (< 0.1% per sample, cf Ogier, J.-C., Pagès, S., Galan, M., Barret, M., & Gaudriault, S. (2019). rpoB, a promising marker for analyzing the diversity of bacterial communities by amplicon sequencing. BMC Microbiology, 19(1), 171. https://doi.org/10.1186/s12866-019-1546-z)

For other communities, done with *taxize* package :
# cleaning step 1 = *taxize::gnr_resolve* and small manual corrections
# cleaning step 2 = *taxize::tax_name*
# cleaning step 3 = manual additions
tax_community_1 = object to use for taxize input (query)
tax_community_2 = raw taxize output (ITIS and NCBI results)

For all communities :
tax_community_3 = cleaned taxonomy = ***tax_table*** by community
tax_ALL = all communities combined = ***tax_table*** with all communities

Replace community by :

- macrofauna_surface
- macrofauna_aerial
- macrofauna_foliar
- nematodes
- micro_arthropodes
- microorganisms
  - bacteria
  - fungi
  - protists

Some details of the steps

- step C = functional assignation

community_guilds_taxalevel (taxalevel = species or genus or family or order (order only for microorganisms because the function was to long = the taxa level was too high for the other communities) with the *get.guilds* function)
community_interactions_taxalevel (idem with the *get.interactions* function)

community_guilds (all taxalevels combined) = **fun_table** by community for guilds
community_interactions (idem) = **fun_table** by community for interactions

fun_guilds_ALL = all communities combined = ***fun_table*** with all communities for guilds
fun_interactions_ALL = all communities combined = ***fun_table*** with all communities for interactions

Gratin functions for assignations :

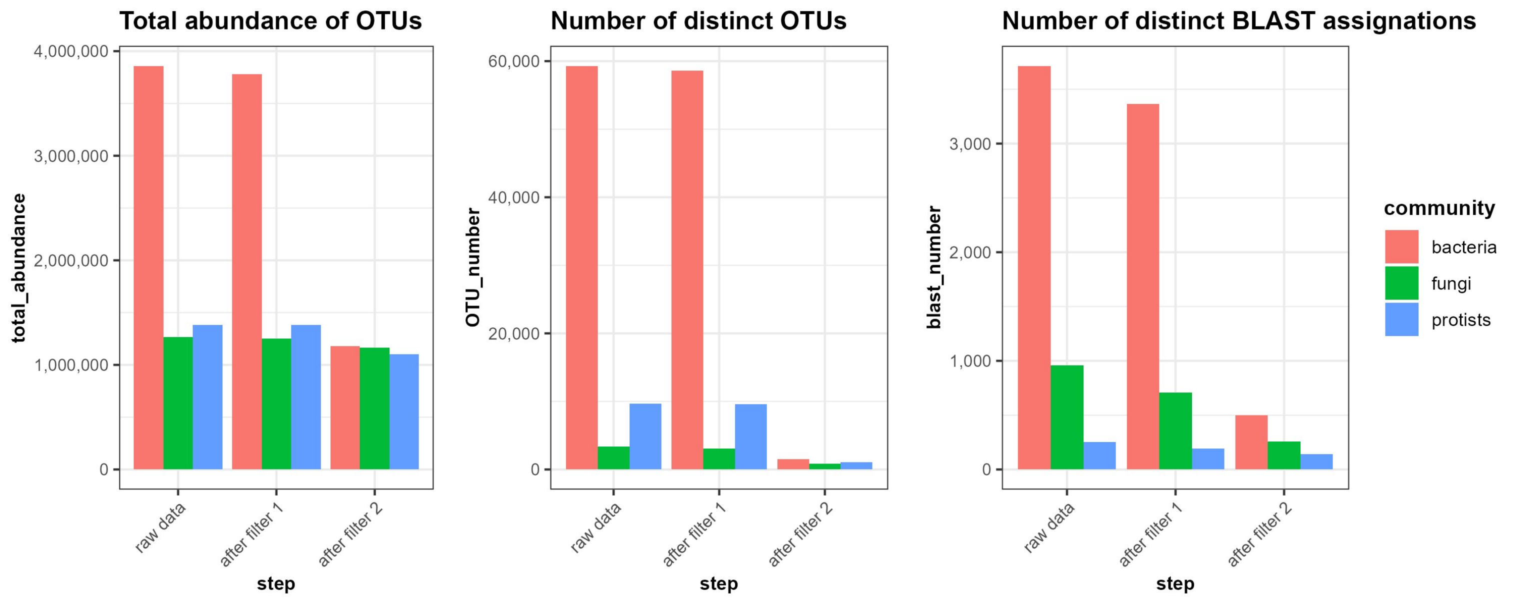
- *get.guilds*
- *get.interactions*
- *get.trophic.groups* (not used here)
- *get.diets* (not used here)

Replace community by :

- macrofauna_surface
- macrofauna_aerial
- macrofauna_foliar
- nematodes
- micro_arthropodes
- microorganisms
    - bacteria
    - fungi
    - protists

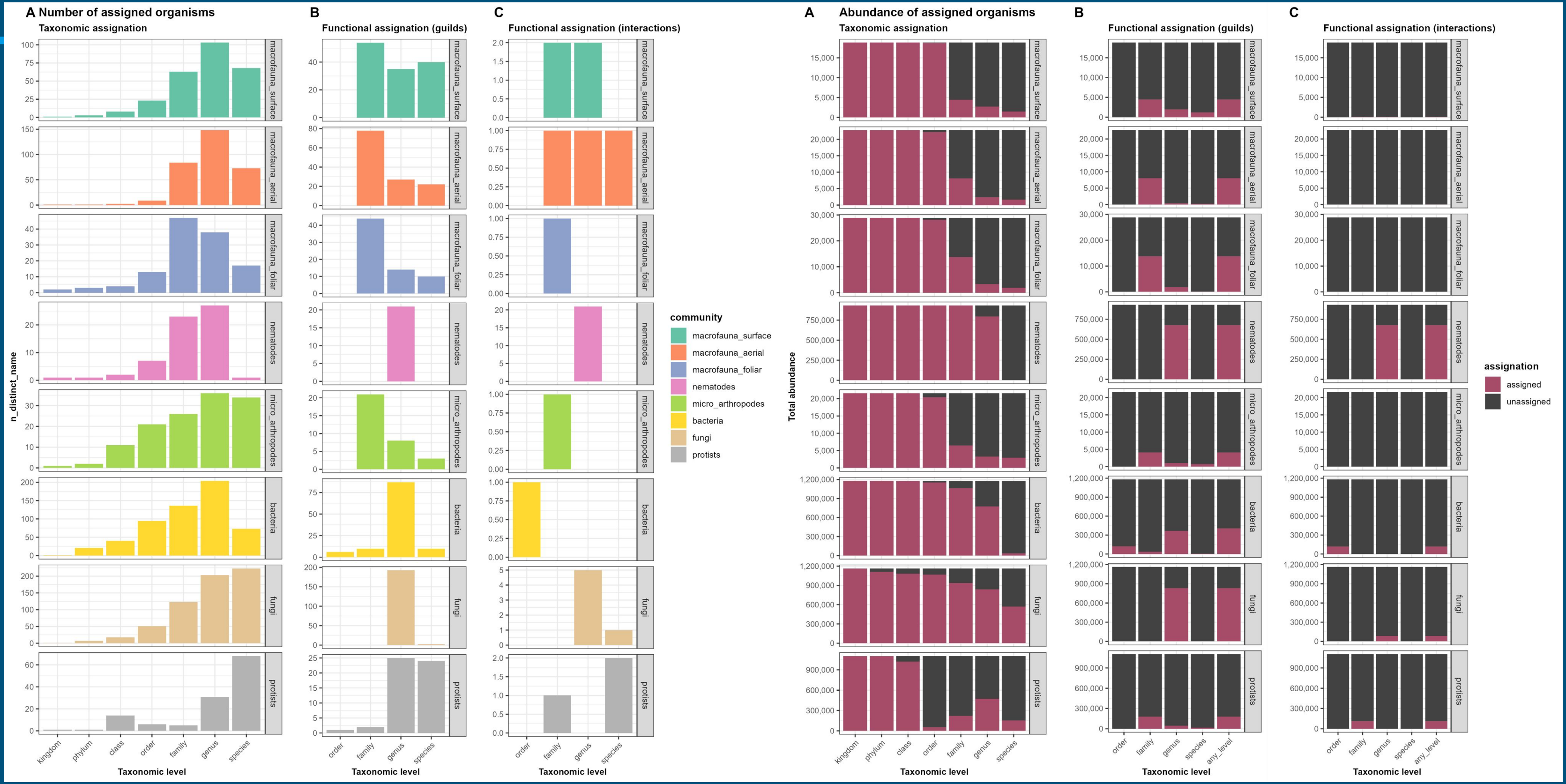
Tracking outputs

data = metabarcoding



Tracking outputs

data = all dates available



Some details of the steps

- step D = merge into *phyloseq* objects

Prepare constructors for all communities :
tidy_data_PO (*otu_table*)
metadata_PO (*sam_data*)
fun_table_guilds_PO (to join in *tax_table*)
fun_table_inter_PO (to join in *tax_table*)

Then merge into PO for each community :
tidy_data_PO_community
tax_table_PO_community
metadata_PO_community

Replace community by :

- macrofauna_surface
- macrofauna_aerial
- macrofauna_foliar
- nematodes
- micro_arthropodes
- microorganisms
  - bacteria
  - fungi
  - protists

PO_community_raw <- phyloseq(tidy_data_PO_community, tax_table_PO_community, metadata_PO_community)
PO_community_stand <- standardize abundances to the median sequencing depth (ref = https://joey711.github.io/phyloseq/preprocess.html + McKnight, D. T., Huerlimann, R., Bower, D. S., Schwarzkopf, L., Alford, R. A., & Zenger, K. R. (2019). Methods for normalizing microbiome data : An ecological perspective. Methods in Ecology and Evolution, 10(3), 389-400. https://doi.org/10.1111/2041-210X.13115)

```
> total = median(sample_sums(PO_community))
> standf = function(x, t=total) round(t*(x/sum(x)))
> PO_community_stand = transform_sample_counts(PO_community,  standf)
```

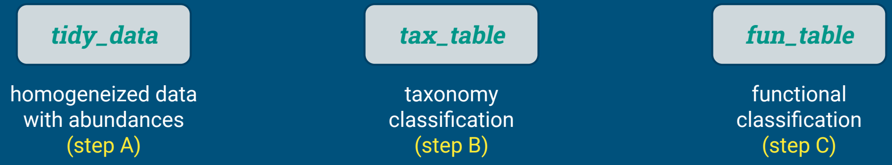
PO_community_stand_prop <- PO_community_stand transformed via proportions
PO_community_norm <- normalize via proportions (ref = https://adrientaudiere.github.io/MiscMetabar/reference/normalize_prop_pq.html)

```
> PO_community_norm <- normalize_prop_pq(PO_community, base_log = 2, digits = 0)
```

PO_community_norm_prop <- PO_community_norm transformed via proportions

Then merge PO by versions :
PO_all_stand <- merge_phyloseq(PO_community1_stand, PO_community2_stand...) # used for alpha diversity
PO_all_stand_prop <- merge_phyloseq(PO_community1_stand_prop, PO_community2_stand_prop...) # used for beta diversity
PO_all_norm_prop <- merge_phyloseq(PO_community1_norm_prop, PO_community2_norm_prop...) # used for abundance graphs and tests

Flowchart



tidy_data

homogeneized data
with abundances
(step A)

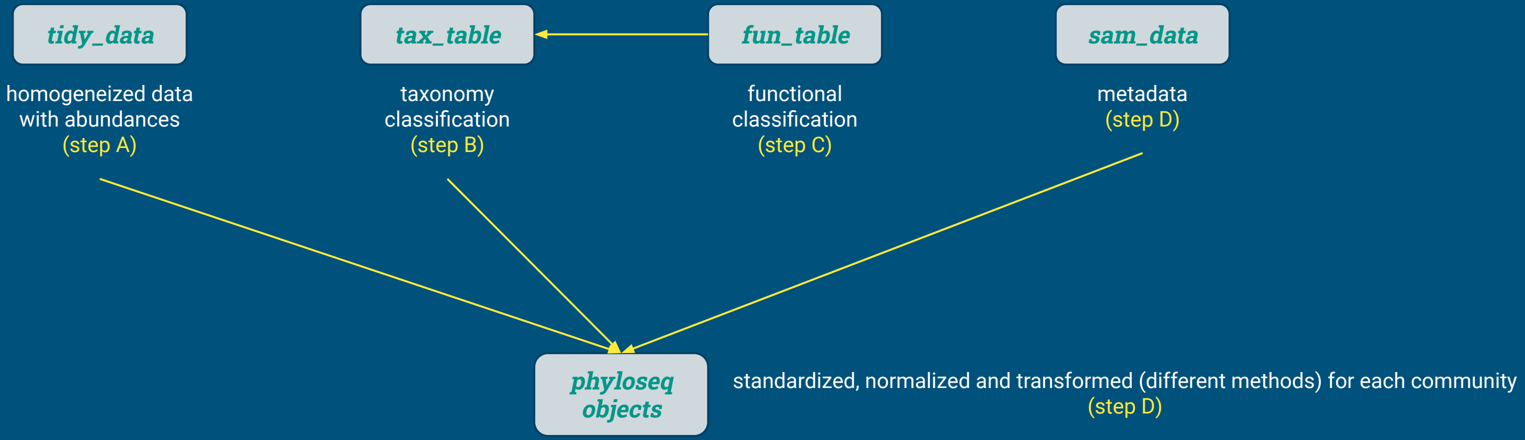
tax_table

taxonomy
classification
(step B)

fun_table

functional
classification
(step C)

Flowchart



tidy_data

homogeneized data
with abundances
(step A)

tax_table

taxonomy
classification
(step B)

fun_table

functional
classification
(step C)

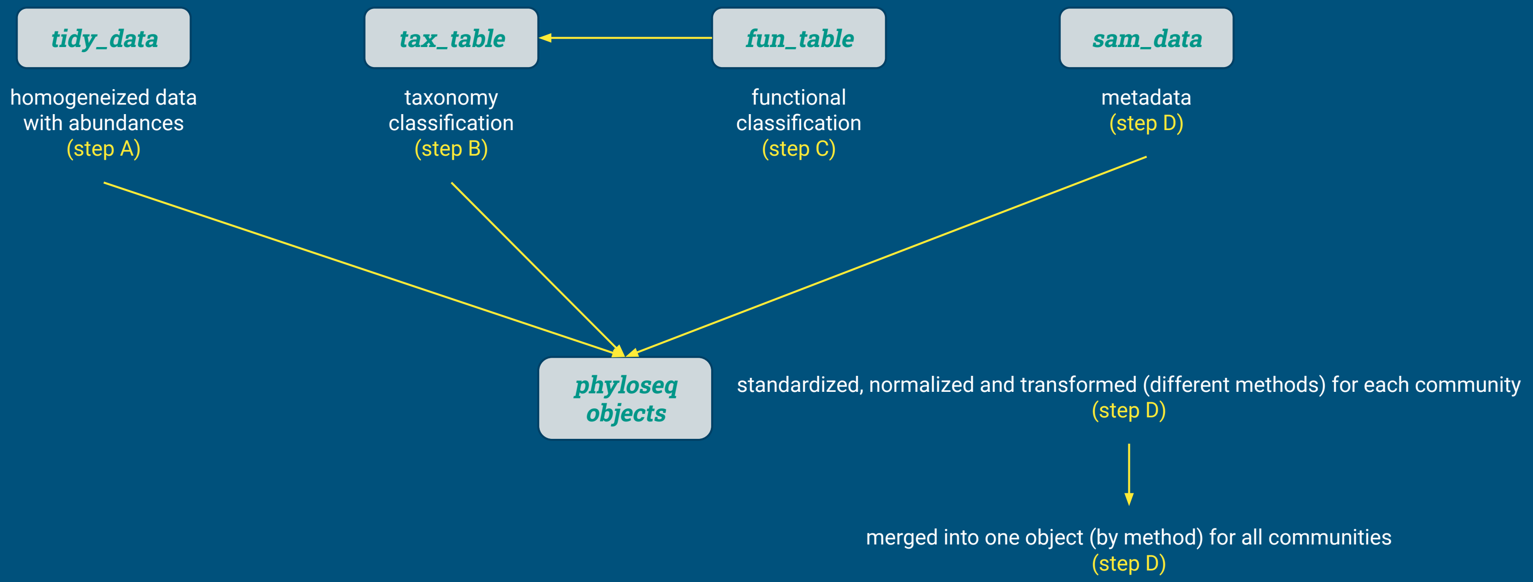
sam_data

metadata
(step D)

phyloseq objects

standardized, normalized and transformed (different methods) for each community
(step D)

Flowchart



tidy_data

homogeneized data
with abundances
(step A)

tax_table

taxonomy
classification
(step B)

fun_table

functional
classification
(step C)

sam_data

metadata
(step D)

phyloseq objects

standardized, normalized and transformed (different methods) for each community
(step D)

merged into one object (by method) for all communities
(step D)

Architecture GitHub

https://github.com/AnneSoMasson/IFOSSA-anneso/tree/main

- root
    - data
        - raw_data
        - derived_data
            - tidy_data
            - tax_table
                - intermediate_files
                - final_files
            - fun_table
            - phyloseq_objects
    - analyses
        - tracking
            - abundance_taxa
            - lists_assignations
            - number_groups
            - number_taxa
            - percentage_abundance
            - percentage_number
        - preanalyses
    - manual