Tutorial for Bioregionalization R package

`virtual_sp` is a dataset simulated that comes with the package. This dataset relies on the response curve of virtual species to a virtual raster. The virtual raster contains 10000 cells and was simulated using `gstat` R package. See here for details.

Based on this layer, the `virtualspecies` R package (Leroy et al. 2015) was used to simulate the response curve of 100 virtual species. A Gaussian curve was used. The mean and standard deviation of the response function was varying among species, such as some of them are more or less generalists/specialists.

For every species in every cell, we could derive a suitability index. Species with suitability index inferior to 0.15 were arbitrarily set absent.

```
# Import virtual dataset
data("virtual_sp")
```

The first step is to convert the data.frame into a contingency table.

```
sp_mat <- contingency(sp_df, "sp", "site", "pa", binary = TRUE)
```

We then need to project the network.

```
sp_proj <- project_network(sp_mat, similarity = "simpson")
sp_proj <- sp_proj[, c("id1", "id2", "simpson")]

# write.table(sp_proj[, c("id1", "id2", "simpson")], "OSLOM2/vignette.txt",
#             row.names = FALSE)
```

Running OSLOM (under Linux distribution).

Here: ./oslom_undir -r 10 -seed 1000 -t 0.5 -cp 0.5 -f vignette.txt -w

Converting the OSLOM .tp file into a list.

```
res <- readLines("../OSLOM2/vignette.txt_oslo_files/tp")
oslom_vignette <- oslom_output(res, sp_mat)
length(unique(oslom_vignette$bioregion))
```

```
## [1] 2
```

Function to compute zscores. Step 3 of Figure 2 in Lenormand et al. (2019)

$$\rho_{ij} = \frac{n_{ij} - \frac{n_i n_j}{n}}{\sqrt{(\frac{n-n_j}{n-1}(1-\frac{n_j}{n})\frac{n_j n_i}{n})}}$$

```
tmp <- left_join(sp_df, oslom_vignette, by = "site")
z_scores <- zscore(tmp, sp_col = "sp", site_col = "site",
                   bioregion_col = "bioregion")

# top10 <- zscore_sp %>%
#   group_by(bioregion_type, bioregion_value) %>%
```

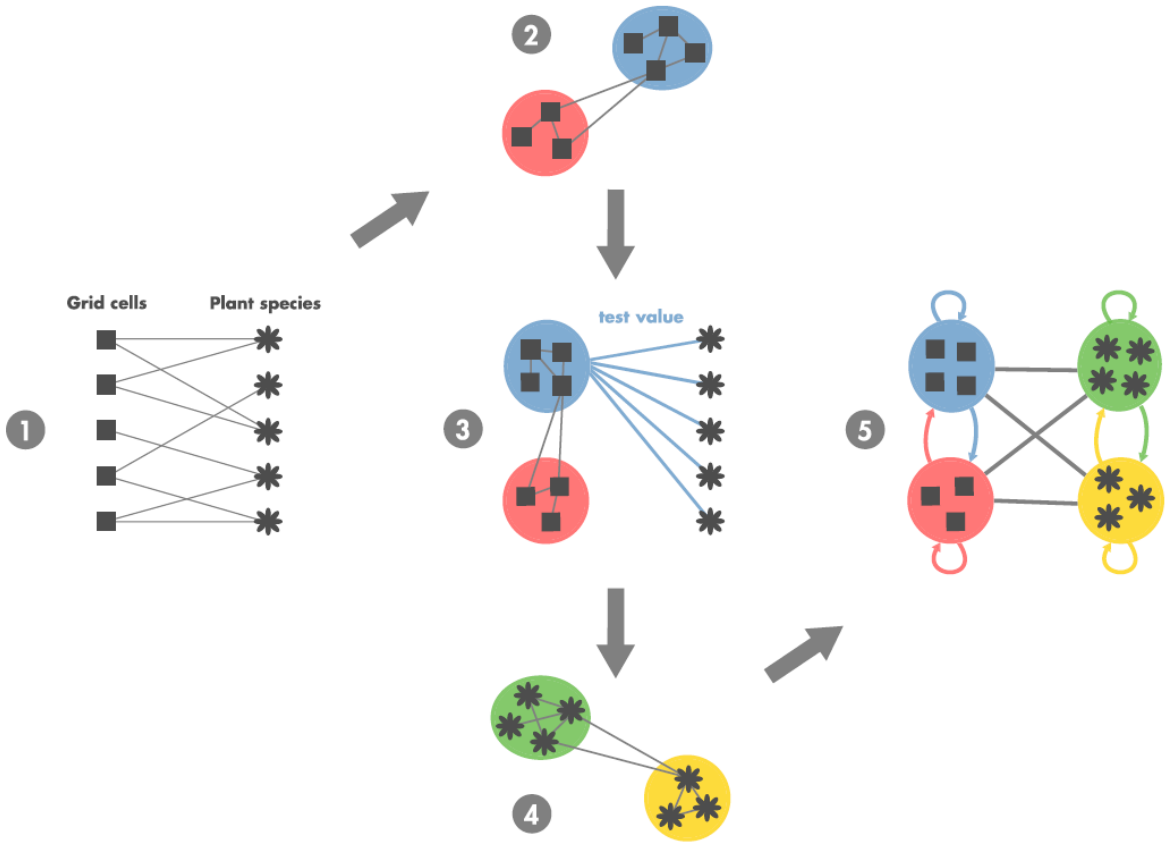


Figure 1: Steps of the biogeographical network analysis. 1. Biogeographical bipartite network where grid cells and species are linked by the presence of a species (or a group of species) in a given grid cell during a certain time window. Note that there is no link between nodes belonging to the same set. 2. The bipartite network is then spatially projected by using a similarity measure of species composition between grid cells. Bioregions are then identified with a network community detection algorithm. 3. The test value matrix based on the contribution of species to bioregions is computed. 4. Then, a network of similarity between species is built, based on the test value matrix. Groups of species sharing similar spatial features are identified using a community detection algorithm. 5. Finally, a coarse-grained biogeographical network unveiling the biogeographical structure of the studied area and the relationship between bioregions is obtained.

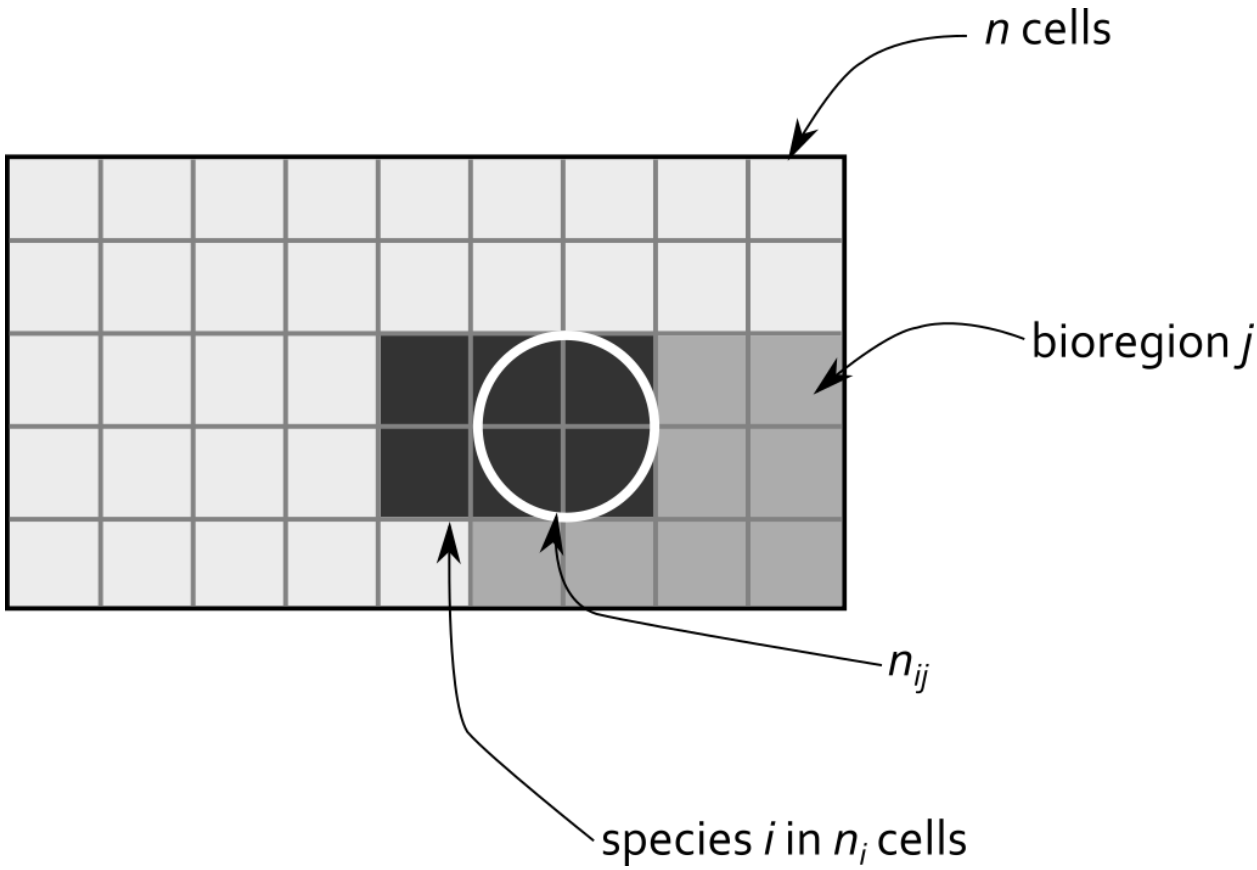


Figure 2: Principle of the zscore calculation.

```
#   mutate(zscore = 100*(zscore-min(zscore))/ # convert zscore into percentages
#            (max(zscore) - min(zscore))) %>%
#   filter(n_i > quantile(n_i, 0.25)) %>% # remove 25% rarest species
#   top_n(10, zscore) %>% # extract top 10
#   mutate(rank = rank(-zscore, # ranking zcore in an ascending order
#                      ties.method = "first")) %>% # if tie zscore, first species
#   select(sp, bioregion_type, bioregion_value, zscore, rank) %>%
#   mutate(zscore = round(zscore, 1)) %>% # rounding zscore to 1 digit
#   as.data.frame()
#
# top10_sp <- top10 %>%
#   select(-zscore) %>% # remove zscore column
#   group_by(bioregion_type, bioregion_value) %>%
#   gather(key = sp_name, value = sp, -bioregion_type, -bioregion_value, -rank) %>%
#   unite(sp_rank, sp_name, rank) %>%
#   spread(sp_rank, sp) %>%
#   as.data.frame()
```

Interaction plots.

```
ex_lambda <- lambda(dat = z_scores, sp_col = "sp", zscore_col = "zscore",
                    bioregion_col = "bioregion",
                    criterion = "top10", plot = TRUE)
```

Example with Ward analysis and k-means clustering.

```
# CA_res <- CA_cluster(sp_mat)
ward_res <- ward_cluster(sp_mat)
```

Projection on a map.

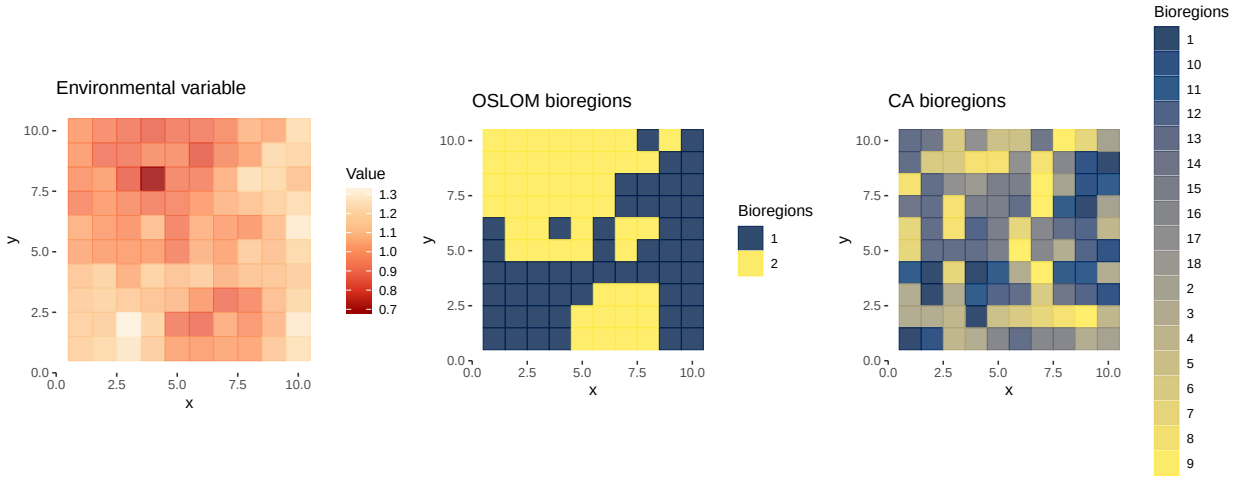
```
plot_grid(
  # Plot of environmental values
  sp_df %>%
    distinct(site, .keep_all = TRUE) %>%
    ggplot(aes(x, y)) +
    geom_tile(aes(fill = env, color = env),
              alpha = 0.8, width = 1, height = 1) +
    scale_color_distiller("Value", palette = "OrRd") +
    scale_fill_distiller("Value", palette = "OrRd") +
    coord_equal() +
    labs(title = "Environmental variable") +
    theme(panel.background = element_rect(fill = "transparent",colour = NA)),

  # Plot of OSLOM bioregions
  sp_df %>%
    left_join(oslom_vignette, by = "site") %>%
    distinct(site, .keep_all = TRUE) %>%
    ggplot(aes(x, y)) +
    geom_tile(aes(fill = as.factor(bioregion), color = as.factor(bioregion)),
              alpha = 0.8, width = 1, height = 1) +
    scale_color_viridis_d("Bioregions", option = "E") +
```

```
  scale_fill_viridis_d("Bioregions", option = "E") +
  coord_equal() +
  labs(title = "OSLOM bioregions") +
  theme(panel.background = element_rect(fill = "transparent",colour = NA)),

# Plot of Ward bioregions
sp_df %>%
  left_join(ward_res, by = "site") %>%
  distinct(site, .keep_all = TRUE) %>%
  ggplot(aes(x, y)) +
  geom_tile(aes(fill = as.factor(cluster), color = as.factor(cluster)),
            alpha = 0.8, width = 1, height = 1) +
  scale_color_viridis_d("Bioregions", option = "E") +
  scale_fill_viridis_d("Bioregions", option = "E") +
  coord_equal() +
  labs(title = "CA bioregions") +
  theme(panel.background = element_rect(fill = "transparent",colour = NA)),

nrow = 1)
```




```
# With sf
# res_map <- sp_df %>%
#   left_join(oslom_vignette, by = "site") %>%
#   st_as_sf(coords = c("x", "y")) %>%
#   group_by(bioregion) %>%
#   summarise()
#
# ggplot(res_map) +
#   geom_sf(aes(fill = as.factor(bioregion), color = as.factor(bioregion))) +
#   scale_fill_viridis_d("Bioregions") +
#   scale_color_viridis_d("Bioregions") +
#   labs(x = "Longitude", y = "Latitude")
```