# Tutorial for Bioregionalization R package

*Pierre Denelle*

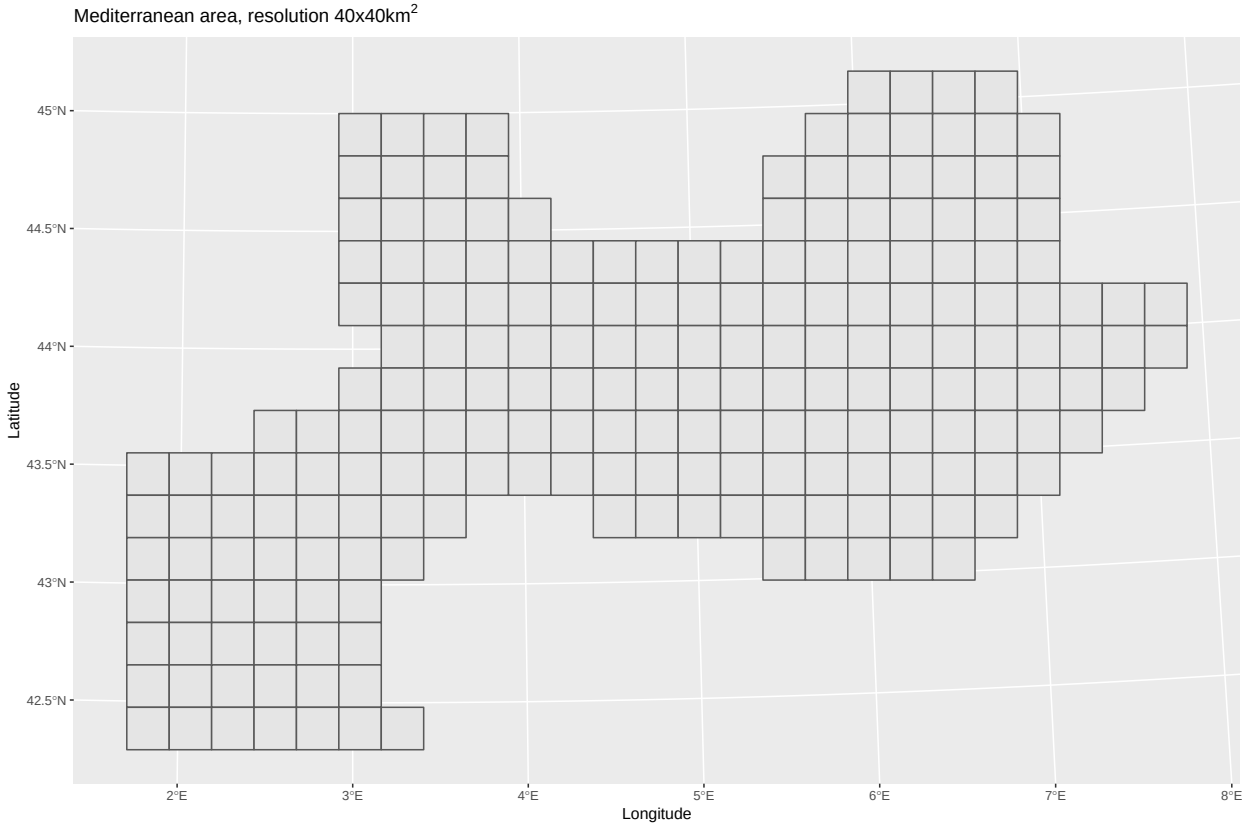
*2019-10-28*

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

# 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 Mediterranean area
medit[[2]] %>%
  ggplot() +
  geom_sf() +
  labs(title = expression("Mediterranean area, resolution 40x40km"^2),
       x = "Longitude", y = "Latitude")
```




```
sp_df <- medit[[1]] %>%
  rename(site = id) %>%
  mutate(site = as.character(site),
         sp = as.character(sp))

colnames(medit[[2]]) <- c("site", "geometry")
medit[[2]]$site <- as.character(medit[[2]]$site)
colnames(medit[[2]]) <- c("site", "geometry")
```

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

```
#sp_mat <- contingency(sp_df, "site", "sp", "pa", binary = TRUE)
sp_mat <- contingency(sp_df, "site", "sp", ab = NULL, binary = TRUE)
knitr::kable(sp_mat[1:5, 1:5])
```

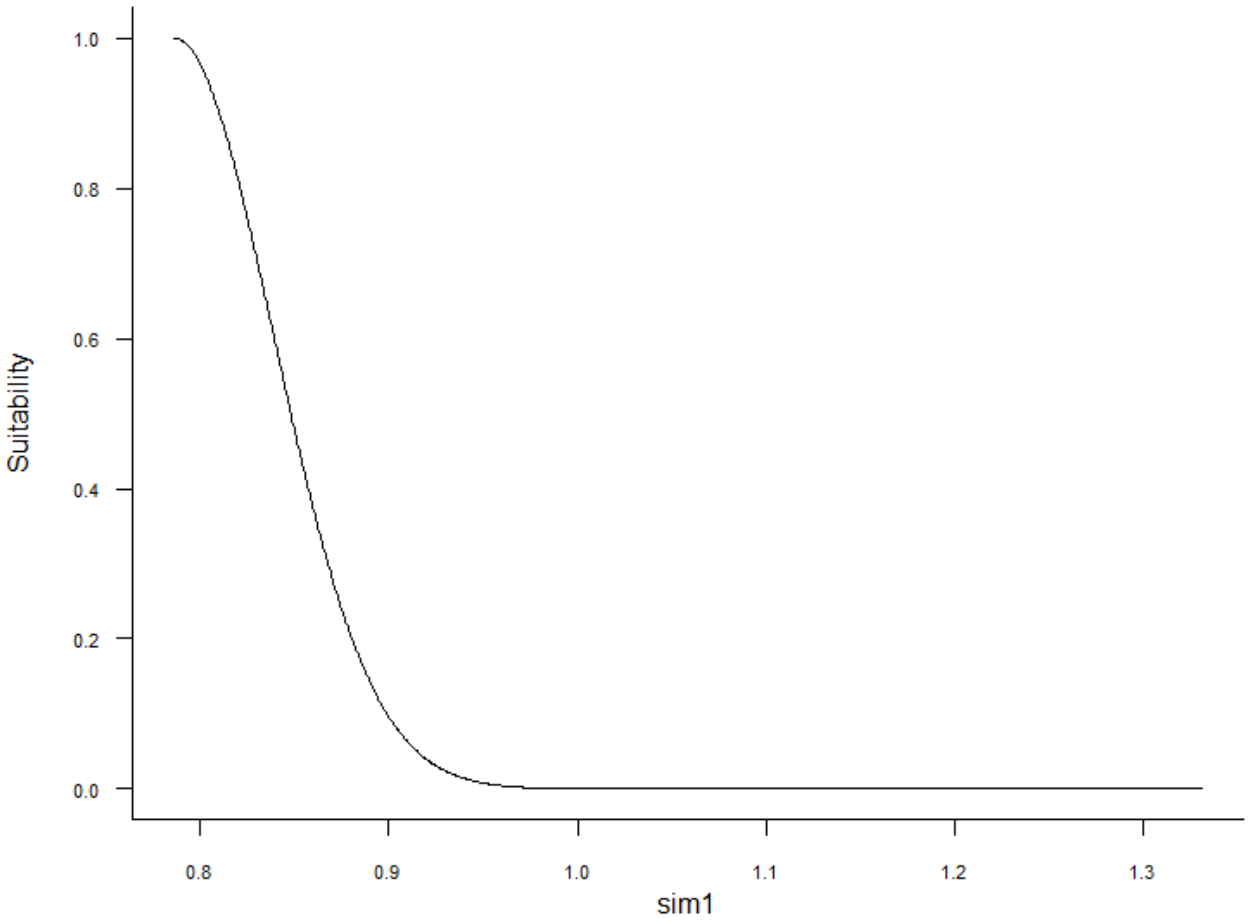



Figure 1: Example of response curve for one virtual species.

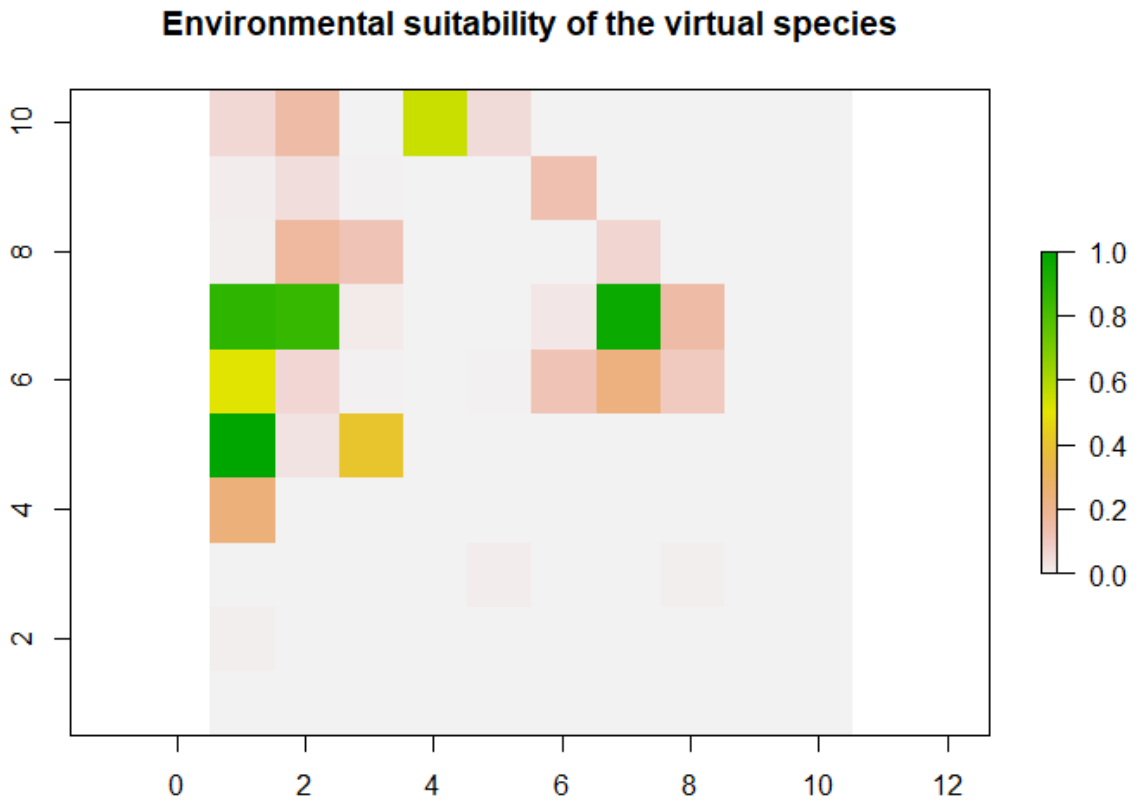



Figure 2: Example of suitability map for one virtual species.

| | 100787 | 101188 | 103246 | 103316 | 103375 |
|---|---|---|---|---|---|
| 1030.23076923077 | 0 | 0 | 0 | 0 | 0 |
| 1113 | 0 | 0 | 0 | 0 | 0 |
| 1117 | 0 | 0 | 0 | 0 | 0 |
| 1119.5 | 0 | 0 | 0 | 0 | 0 |
| 1119.85714285714 | 0 | 0 | 0 | 0 | 0 |

We then need to project the network.

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

knitr::kable(head(sp_proj))
```

| id1 | id2 | simpson |
|---|---|---|
| 2 | 3 | 1 |
| 2 | 4 | 1 |
| 3 | 4 | 1 |
| 7 | 8 | 1 |
| 12 | 13 | 1 |
| 14 | 18 | 1 |

Running OSLOM.

```
run_oslom(sp_proj, n_runs = 5, t_param = 0.1, cp_param = 0.5,
          saving_directory = "D:/PIERRE_DENELLE/CarHab/Bioregionalization_extra/")
```

Converting the OSLOM .tp file into a list.

```
#res <- readLines("../OSLOM2/vignette.txt_oslo_files/tp")
res <- readRDS("../../Bioregionalization_extra/tp.rds")
oslom_vignette <- oslom_output(res, sp_mat)

print(paste0("Number of bioregions detected = ",
             length(unique(oslom_vignette$bioregion))))
```

```
## [1] "Number of bioregions detected = 24"
```

Step 3 of Figure 1 (see Lenormand et al. (2019))

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

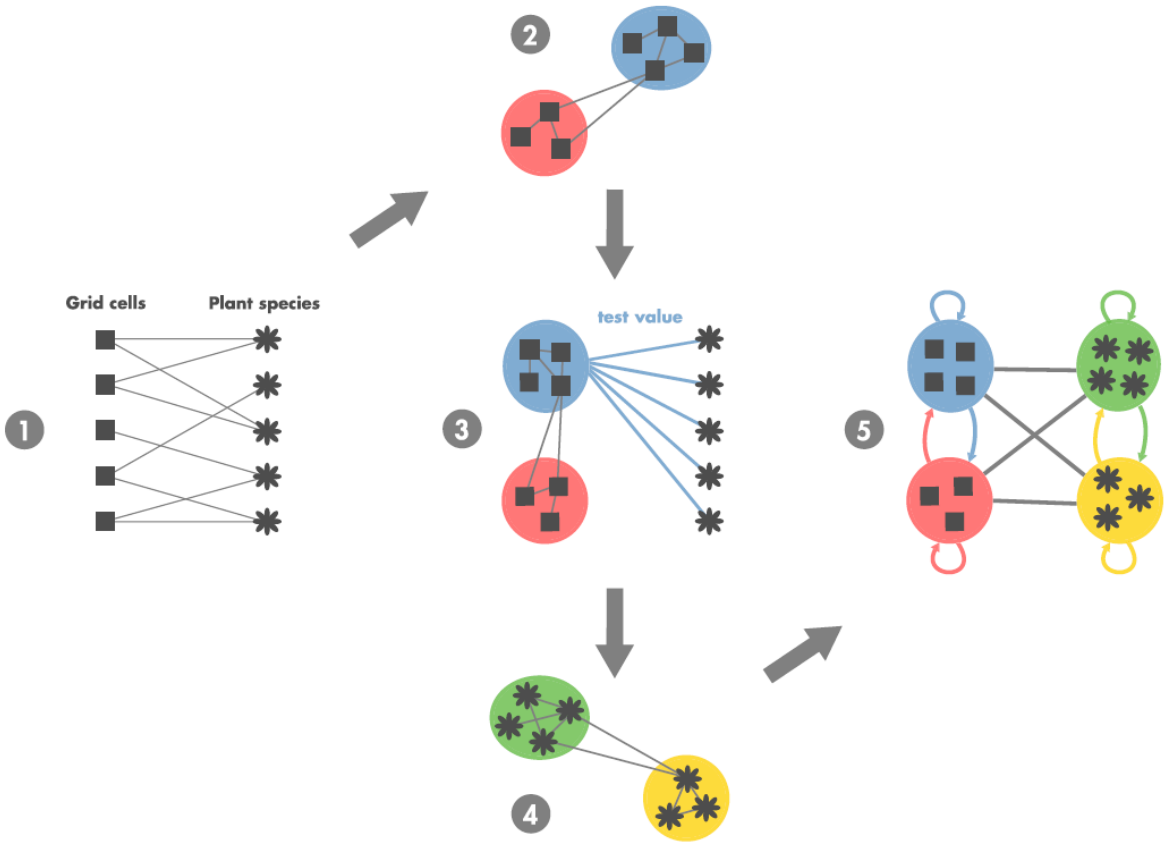



Figure 3: Steps of the biogeographical network analysis.1.Biogeographical bipartite network where grid cells and species are linked by the presence of a species (or a group of species) in a given grid cell during a certain time window. Note that there is no link between nodes belonging to the same set. 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.

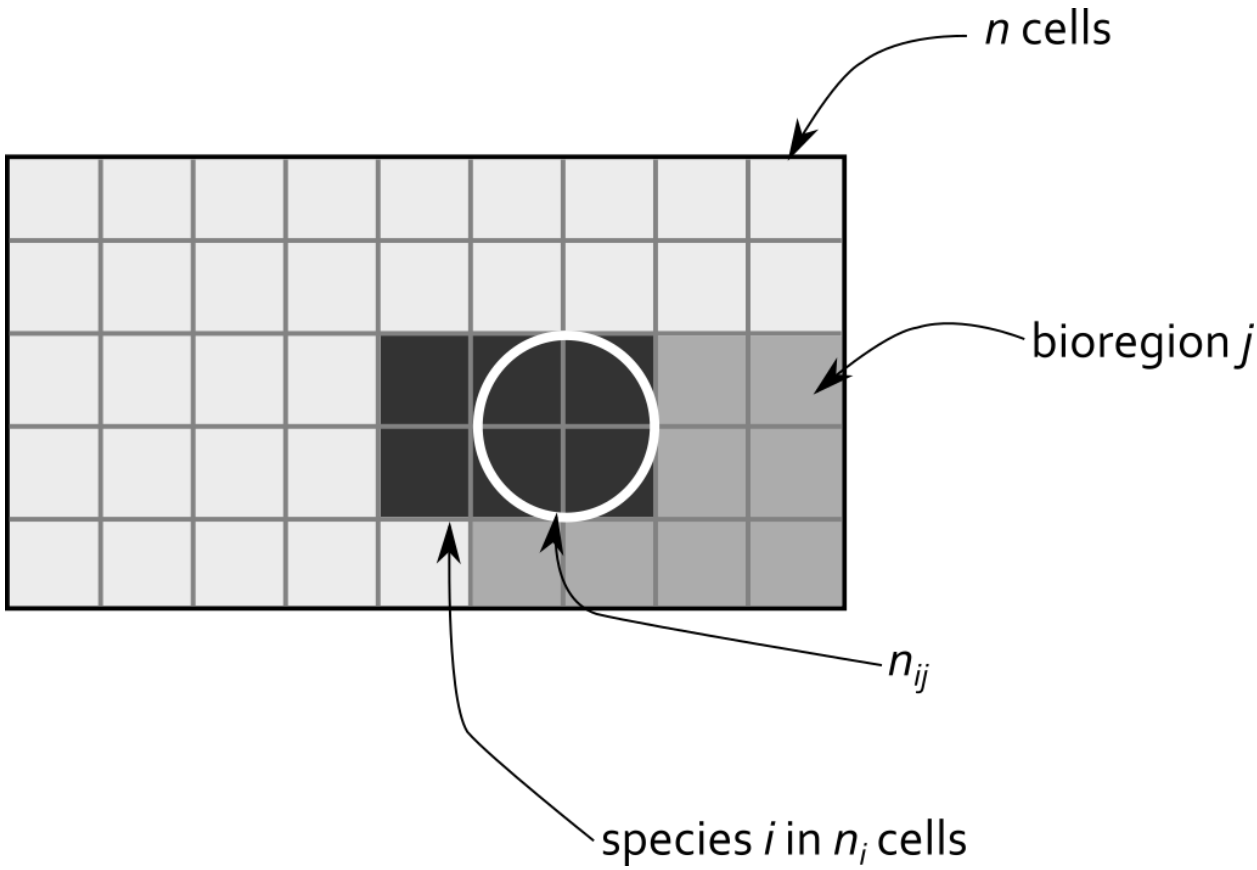



Figure 4: Principle of the zscore calculation.

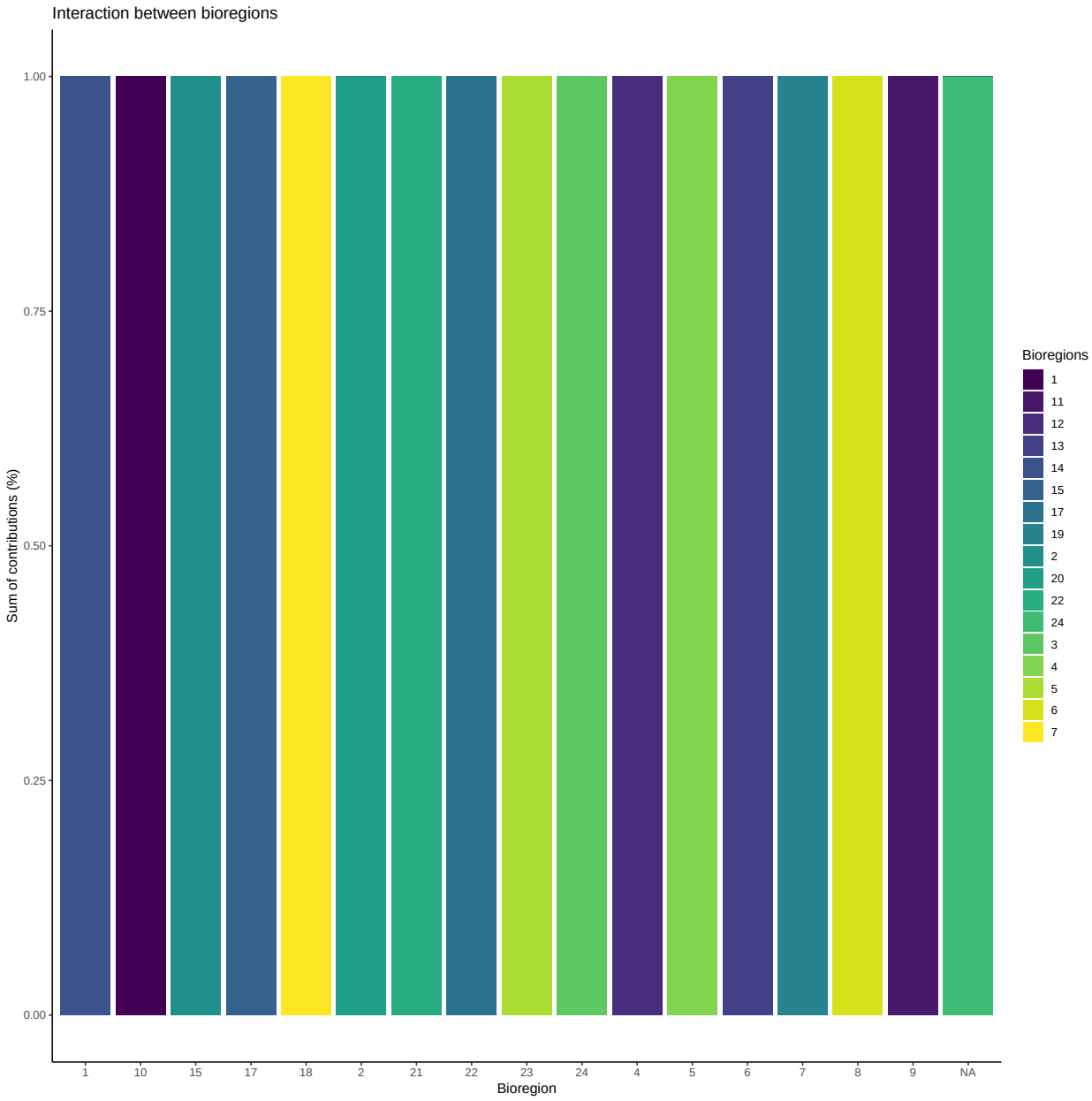
```r
# tmp <- left_join(sp_df, oslom_vignette, by = "site")
tmp <- left_join(sp_df, oslom_vignette, by = "site")
z_scores <- zscore(tmp, sp_col = "sp", site_col = "site",
                   bioregion_col = "bioregion")

top10 <- z_scores %>%
  group_by(bioregion) %>%
  top_n(n = 10, zscore) %>% # extract top 10
  mutate(rank = rank(-zscore, # ranking zcore in an ascending order
                     ties.method = "first")) %>% # if tie zscore, first species
  dplyr::select(sp, bioregion, zscore, rank) %>%
  mutate(zscore = round(zscore, 1)) %>% # rounding zscore to 1 digit
  as.data.frame()

knitr::kable(top10[which(top10$bioregion == "2"), ])
```

| | sp | bioregion | zscore | rank |
|---|---|---|---|---|
| 67 | 84264 | 2 | 41.8 | 1 |
| 68 | 84472 | 2 | 14.5 | 2 |

Interaction plots.

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

ex_lambda
```

```
## [[1]]
##    focal_bioregion bioregion sum_rho
## 1               21        22       1
## 2               22        17       1
## 3               23         5       1
## 4               17        15       1
## 5             <NA>        17       0
## 6             <NA>        24       1
## 7                5         4       1
## 8               18         7       1
## 9                8         6       1
## 10              15         2       1
## 11              10         1       1
## 12              24         3       1
## 13               9        11       1
## 14               4        12       1
## 15               7        19       1
## 16               6        13       1
## 17               2        20       1
## 18               2        19       0
## 19               1        14       1
##
## [[2]]
```




Example with Ward analysis and k-means clustering.

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

Bipartite algorithms applied on example dataset.

```
# With fastgreedy
bip <- algo_bipartite(dat = sp_mat, algo = "greedy", weight = FALSE)

# With Beckett algorithm
bip2 <- algo_bipartite(dat = sp_mat, algo = "LPAwb", weight = FALSE)

# Only sites
```

```r
bip_site <- bip %>%
  filter(cat == "site") %>%
  rename(site = node) %>%
  dplyr::select(site, module)

bip_site2 <- bip2 %>%
  filter(cat == "site") %>%
  rename(site = node) %>%
  dplyr::select(site, module)
```

Cz computation on bipartite results.

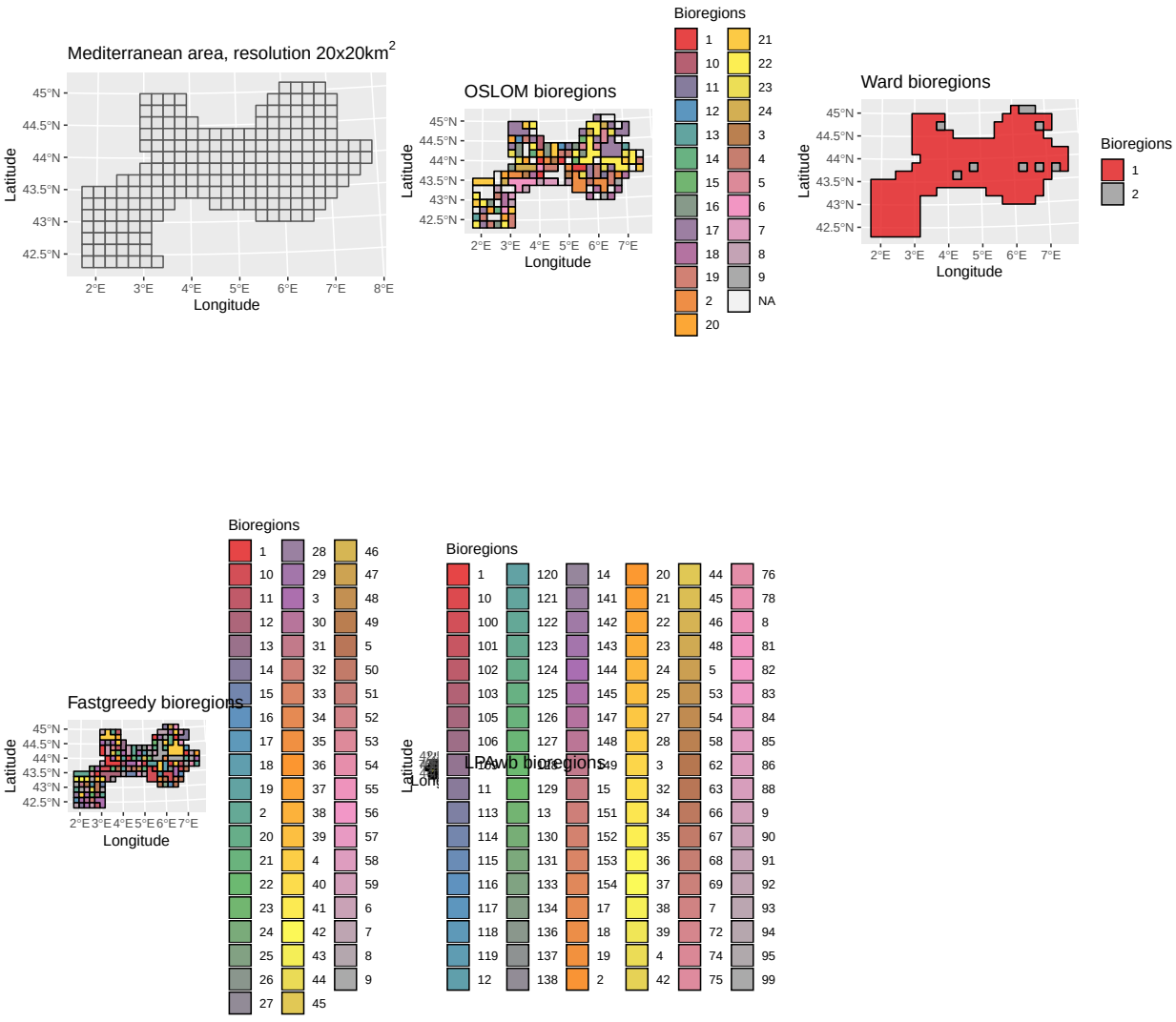
```r
bip_cz <- bip[, c("node", "module", "cat")]
colnames(bip_cz) <- c("node", "mod", "cat")

link_cz <- sp_df[, c("site", "sp")] %>%
  left_join(bip_cz[which(bip_cz$cat == "site"), c("node", "mod")],
            by = c("site" = "node")) %>%
  rename(mod_site = mod) %>%
  left_join(bip_cz[which(bip_cz$cat == "sp"), c("node", "mod")],
            by = c("sp" = "node")) %>%
  rename(mod_sp = mod)

cz_bip <- cz(link_dat = link_cz, dat = bip_cz, ab = NULL)

head(cz_bip[[1]])
```

```
##      node mod  cat C n_link_mod mean_link_mod sd_link_mod          z
## 1    1113   1 site 0         10      9.565217     10.2373 0.04247043
## 2    1117   1 site 0         10      9.565217     10.2373 0.04247043
## 3  1119.5   1 site 0         10      9.565217     10.2373 0.04247043
## 4    1157   1 site 0         10      9.565217     10.2373 0.04247043
## 5 1340.75   1 site 0         10      9.565217     10.2373 0.04247043
## 6 1434.75   1 site 0         10      9.565217     10.2373 0.04247043
```

Projection on a map.

```r
color_vector <- c(brewer.pal(12, "Paired"), brewer.pal(12, "Set3"))
getPalette <- colorRampPalette(brewer.pal(9, "Set1"))

plot_grid(
  # Plot of Mediterreanean region
  medit[[2]] %>%
    ggplot() +
    geom_sf() +
    labs(title = expression("Mediterranean area, resolution 20x20km"^2),
         x = "Longitude", y = "Latitude"),
  # Plot of OSLOM bioregions
  sp_df %>%
    left_join(oslom_vignette, by = "site") %>%
    distinct(site, .keep_all = TRUE) %>%
    left_join(medit[[2]], by = "site") %>%
```

```
  st_as_sf() %>%
  group_by(bioregion) %>%
  summarise() %>%
  ggplot() +
  geom_sf(aes(fill = as.factor(bioregion)), color = "black",  alpha = 0.8) +
  scale_fill_manual("Bioregions",
                    values = getPalette(length(unique(oslom_vignette$bioregion)))) +
  labs(title = "OSLOM bioregions", x = "Longitude", y = "Latitude"),

# Plot of Ward bioregions
sp_df %>%
  left_join(ward_res, by = "site") %>%
  distinct(site, .keep_all = TRUE) %>%
  left_join(medit[[2]], by = "site") %>%
  st_as_sf() %>%
  group_by(cluster) %>%
  summarise() %>%
  ggplot() +
  geom_sf(aes(fill = as.factor(cluster)), color = "black", alpha = 0.8) +
  scale_fill_manual("Bioregions",
                    values = getPalette(length(unique(ward_res$cluster)))) +
  labs(title = "Ward bioregions", x = "Longitude", y = "Latitude"),

# Plot of fastgreedy bioregions
sp_df %>%
  left_join(bip_site, by = "site") %>%
  distinct(site, .keep_all = TRUE) %>%
  left_join(medit[[2]], by = "site") %>%
  st_as_sf() %>%
  group_by(module) %>%
  summarise() %>%
  ggplot() +
  geom_sf(aes(fill = as.factor(module)), color = "black", alpha = 0.8) +
  scale_fill_manual("Bioregions",
                    values = getPalette(length(unique(bip_site$module)))) +
  labs(title = "Fastgreedy bioregions", x = "Longitude", y = "Latitude"),

# Plot of LPAwb bioregions
sp_df %>%
  left_join(bip_site2, by = "site") %>%
  distinct(site, .keep_all = TRUE) %>%
  left_join(medit[[2]], by = "site") %>%
  st_as_sf() %>%
  group_by(module) %>%
  summarise() %>%
  ggplot() +
  geom_sf(aes(fill = as.factor(module)), color = "black", alpha = 0.8) +
  scale_fill_manual("Bioregions",
                    values = getPalette(length(unique(bip_site2$module)))) +
  labs(title = "LPAwb bioregions", x = "Longitude", y = "Latitude"),
nrow = 2)
```


Mediterranean area, resolution 20x20km2
OSLOM bioregions
Ward bioregions
Bioregions
1
2
Latitude
Longitude
45°N
44.5°N
44°N
43.5°N
43°N
42.5°N
2°E
3°E
4°E
5°E
6°E
7°E
8°E


Fastgreedy bioregions
LPAwb bioregions
Bioregions
Latitude
Longitude