



```

1 generatePatchySpecies <-
2 function(nb.sim = 1, nb.sp = 20)
3 {
4   richness <- stack()
5   richness.patch <- stack()
6   for (j in 1:nb.sim)
7   {
8     sp.traits <- data.frame(T.optimum = sample(temperature.gradient,
9       nb.sp, replace = T),
10      T.tolerance = sample(seq(50, 100,
11        length = 1000),
12        nb.sp, replace = T))
13     species <- list() # All the species will be stored in this list
14     range.stack <- stack() # Species ranges will be stored in this stack
15     range.patch.stack <- stack() # Species ranges with patched distributions will be stored in this stack
16
17     for (i in 1:nb.sp)
18     {
19       # Step 1: generating the species environmental suitability
20       # (i.e., transforming from E-space to G-space)
21       species[[i]] <- generateSpFromFun(bio1,
22         list(bio1 = list(fun = "gauss.resp",
23           args = list(mean = sp.traits[i, "T.optimum"],
24             diff. = sp.traits[i, "T.tolerance"],
25             prob. = 0.99))),
26         rescale.each.response = FALSE,
27         rescale = FALSE)
28       # Step 2: converting environmental suitability to presence-absence,
29       # using the probabilistic conversion
30       species[[i]] <- convertToPA(species[[i]], # PA.method = "threshold",
31         beta = 0.7, alpha = -0.05,
32         plot = FALSE)
33
34       # Step 2.5: generate habitat patches
35       if(cellStats(species[[i]]$pa.raster, stat = 'max') == 1)
36       {
37         species[[i]]$patched.pa.raster <- raster(matrix(0, nc = 1, nr = 1))
38         count <- 0
39         while(cellStats(species[[i]]$patched.pa.raster, stat = 'max') != 1 | count <= 10)
40         {
41           patches <- generate.patches(bio1, n.patches = 50, patch.size = 10)
42           species[[i]]$patched.pa.raster <- overlay(species[[i]]$pa.raster,
43             patches,
44             fun = function(x, y){return(x * y)})
45           count <- count + 1
46         }
47       } else
48       {
49         species[[i]]$patched.pa.raster <- species[[i]]$pa.raster
50       }
51
52
53
54       # Step 3: we store ranges to calculate richness par cell
55       range.stack <- addLayer(range.stack,
56         species[[i]]$pa.raster)
57       range.patch.stack <- addLayer(range.patch.stack,
58         species[[i]]$patched.pa.raster)
59     }
60
61
62     # Richness calculation
63     richness <- addLayer(richness,
64       sum(range.stack))
65     richness.patch <- addLayer(richness.patch,
66       sum(range.patch.stack))
67
68     # Saving the files
69     save(species, file = paste0("../data/S1_sim", j, "_species"))
70     save(sp.traits, file = paste0("../data/S1_sim", j, "_traits"))
71     writeRaster(range.stack, paste0("../data/S1_sim", j, "_rangestacks"),
72       overwrite = T)
73     writeRaster(richness, paste0("../data/S1_sim", j, "_richness"),
74       overwrite = T)
75     writeRaster(range.patch.stack, paste0("../data/S1_sim", j, "_rangestacks_patch"),
76       overwrite = T)
77     writeRaster(richness.patch, paste0("../data/S1_sim", j, "_richness_patch"),
78       overwrite = T)
79   }
80 }

```

