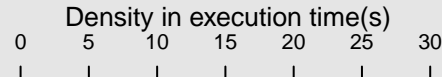


./scripts/07.R



```
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     species <- foreach(i = 1:nb.sp, .export = c("bio1", "generate.patches",
17       "expand")) %dopar%
18     {
19       sup <- custnorm(x = sp.traits[i, "T.optimum"],
20         mean = sp.traits[i, "T.optimum"],
21         diff = sp.traits[i, "T.tolerance"],
22         prob = 0.99)
23
24       cur.sp <- generateSpFromFun(raster.stack = bio1,
25         list(bio1 = list(fun = "custnorm",
26           args = list(mean = sp.traits[i, "T.optimum"],
27             diff = sp.traits[i, "T.tolerance"],
28             prob = 0.99))))
29       cur.sp$suitab.raster <- cur.sp$suitab.raster / sup
30
31       cur.sp <- convertToPA(cur.sp, # PA.method = "threshold",
32         beta = 0.7, alpha = -0.05,
33         plot = FALSE)
34
35       # Step 2.5: generate habitat patches
36       patches <- generate.patches(bio1, n.patches = 50, patch.size = 10)
37       cur.sp$patched.pa.raster <- overlay(cur.sp$pa.raster,
38         patches,
39         fun = function(x, y) x * y)
40
41       return(cur.sp)
42     }
43   }
44   # Richness calculation
45   range.stack <- stack(sapply(species, FUN = function(x) return(x$pa.raster)))
46   range.patch.stack <- stack(sapply(species, FUN = function(x) return(x$patched.pa.raster)))
47   # Try rasterengine here
48   richness <- sum(range.stack)
49   richness.patch <- sum(range.patch.stack)
50
51   # Saving the files
52   save(species, file = paste0("./data/S1_sim", j, "_species"))
53   save(sp.traits, file = paste0("./data/S1_sim", j, "_traits"))
54   writeRaster(range.stack, paste0("./data/S1_sim", j, "_rangestacks"),
55     overwrite = T)
56   writeRaster(richness, paste0("./data/S1_sim", j, "_richness"),
57     overwrite = T)
58   writeRaster(range.patch.stack, paste0("./data/S1_sim", j, "_rangestacks_patch"),
59     overwrite = T)
60   writeRaster(richness.patch, paste0("./data/S1_sim", j, "_richness_patch"),
61     overwrite = T)
62 }
63 }
```