

VoCC_Tutorial

Jorge García Molinos

06.05.2019

This vignette provides the code to reproduce the examples for the R package VoCC as presented in García Molinos et al. (2019). Refer to the paper and the function documentation for details on function options, considerations on the argument choices and interpretation of output.

For this tutorial we need the following packages (if not installed get them first):

```
# devtools::install_github("JorGarMol/VoCC", dependencies = TRUE, build_vignettes = TRUE)
library(VoCC)
#> Warning: package 'raster' was built under R version 3.4.4
#> Warning: package 'sp' was built under R version 3.4.4
library(rgeos)
#> Warning: package 'rgeos' was built under R version 3.4.4
library(rasterVis)
#> Warning: package 'rasterVis' was built under R version 3.4.4
#> Warning: package 'LatticeExtra' was built under R version 3.4.4
library(gridExtra)
#> Warning: package 'gridExtra' was built under R version 3.4.4
library(doParallel)
#> Warning: package 'doParallel' was built under R version 3.4.4
library(foreach)
library(scales)
#> Warning: package 'scales' was built under R version 3.4.4
library(data.table)
#> Warning: package 'data.table' was built under R version 3.4.4
library(mapplots)
#> Warning: package 'mapplots' was built under R version 3.4.4
library(ggplot2)
#> Warning: package 'ggplot2' was built under R version 3.4.4
library(repmis)
#> Warning: package 'repmis' was built under R version 3.4.4
```

We also need a few data sets that can be downloaded into R from the Github Data repository (https://github.com/JorGarMol/VoCC_data_sets) companion to the package. See the repository readme file for description of the data sets.

```
source_data("https://github.com/JorGarMol/VoCC_data_sets/blob/master/Data/HadiSST.rda?raw=true")
#> [1] "HadiSST"
source_data("https://github.com/JorGarMol/VoCC_data_sets/blob/master/Data/EZZs.rda?raw=true")
#> [1] "EEZs"
source_data("https://github.com/JorGarMol/VoCC_data_sets/blob/master/Data/marshift.rda?raw=true")
#> [1] "marshift"
source_data("https://github.com/JorGarMol/VoCC_data_sets/blob/master/Data/traj25.rda?raw=true")
#> [1] "traj25"
```

Example 1: Prediction of biogeographical shifts

We have a look first at the “marshift” global data set containing reported range shifts in marine species corresponding to given periods of time.

```
str(marshift)
#> 'data.frame': 343 obs. of 6 variables:
#> $ lat : num 49.7 53.8 53.8 40 43 ...
#> $ long : num -4.33 5 5 1 -9.3 -1.4 -9.3 -71.7 -71.7 -71.7 ...
#> $ timespan : int 30 40 40 55 35 35 84 39 26 54 ...
#> $ years_data: int 23 40 40 15 4 2 5 2 2 2 ...
#> $ taxa : Factor w/ 12 levels "Benthic algae",...: 6 6 6 3 3 4 5 5 5 ...
#> $ Shift : num 536.1 65.6 95.9 40 10 ...
```

Next, we calculate the gradient- and distance-based velocities (1960-2009), using the HSST data set, from which we will later extract the corresponding values for each observed shift.

```
# monthly to annual averages
r <- sumSeries(HadiSST, p = "1960-01/2009-12", yr0 = "1955-01-01", l = nlayers(HadiSST), fun =
# temporal trend
vt <- tempTrend(r, th = 10)
# spatial gradient
vg <- spatGrad(r, th = 0.0001, projected = FALSE)
# climate velocity
gv <- gVoCC(vt, vg)

# Now the distance-based velocities
# Take 1960-1970 as base period against 2000-2009
r2 <- stack(mean(r[[1:10]]), na.rm = TRUE), mean(r[[41:50]]), na.rm = TRUE))
# prepare the data frame with the necessary variables
clim <- na.omit(data.frame(getValues(r2), cid = 1:ncell(r)))
clim[,c("x","y")] <- xyFromCell(r, clim$cid)
# 1965-2004 (40 yr), 500 km search radius
v <- dVoCC(clim, n = 1, tdiff = 40, method = "Single", climTol = 0.1, geoTol = 500, distfun =
```

Next, we extract the mean velocity estimates for each reported shift by taking the average of all grid cell values within a circle of radius equal to the reported range-shift distance. These are then used to fit the simple linear regression models of observed range shifts against climate velocity. Distance-based velocities are strictly positive by definition, so to compare like with like we change first their sign to negative where present local climates are warmer than their future analogues.

```
# Change sign as needed and create the distance-based velocity raster
ind <- which(r2[[1]][v$focal] > r2[[2]][v$target])
#> Warning in .doExtract(x, i, drop = drop): some indices are invalid (NA
#> returned)
v$velBis <- v$vel
v$velBis[ind] <- v$vel[ind] * -1
# put output in raster format
dv <- raster(gv)
dv[v$focal] <- v$velBis

marshift$GV <- with(marshift, raster::extract(abs(gv[[1]]), cbind(long, lat), buffer = (Shift*
```

```

marshift$DV <- with(marshift, raster::extract(abs(dv), cbind(long, lat), buffer = (Shift* (time
# retrieve the closest marine cell for those centroids falling on land
marshift$GV <- with(marshift, ifelse(is.na(GV), gv[[1]][which.min(replace(distanceFromPoints(gv
marshift$DV <- with(marshift, ifelse(is.na(DV), dv[which.min(replace(distanceFromPoints(dv, cb
# fit the regression models
Mgv <- lm(Shift^(1/4) ~ I((GV*10)^(1/4)), data = marshift, weights=years_data)
Mdv <- lm(Shift^(1/4) ~ I((DV*10)^(1/4)), data = marshift, weights=years_data)
summary(Mgv)
#>
#> Call:
#> lm(formula = Shift^(1/4) ~ I((GV * 10)^(1/4)), data = marshift,
#> weights = years_data)
#>
#> Weighted Residuals:
#>      Min       1Q   Median       3Q      Max
#> -9.8734 -2.3435 -0.7512  1.7206 14.7201
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      0.85065     0.22677   3.751 0.000207 ***
#> I((GV * 10)^(1/4)) 0.65356     0.09333   7.002 1.35e-11 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 3.756 on 340 degrees of freedom
#> (1 observation deleted due to missingness)
#> Multiple R-squared:  0.126, Adjusted R-squared:  0.1235
#> F-statistic: 49.03 on 1 and 340 DF, p-value: 1.351e-11
summary(Mdv)
#>
#> Call:
#> lm(formula = Shift^(1/4) ~ I((DV * 10)^(1/4)), data = marshift,
#> weights = years_data)
#>
#> Weighted Residuals:
#>      Min       1Q   Median       3Q      Max
#> -8.9855 -2.5032 -0.5629  1.4095 14.9400
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)      1.72739     0.09894 17.459 < 2e-16 ***
#> I((DV * 10)^(1/4)) 0.30978     0.04032   7.683 1.68e-13 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 3.709 on 340 degrees of freedom
#> (1 observation deleted due to missingness)
#> Multiple R-squared:  0.1479, Adjusted R-squared:  0.1454
#> F-statistic: 59.02 on 1 and 340 DF, p-value: 1.681e-13

```

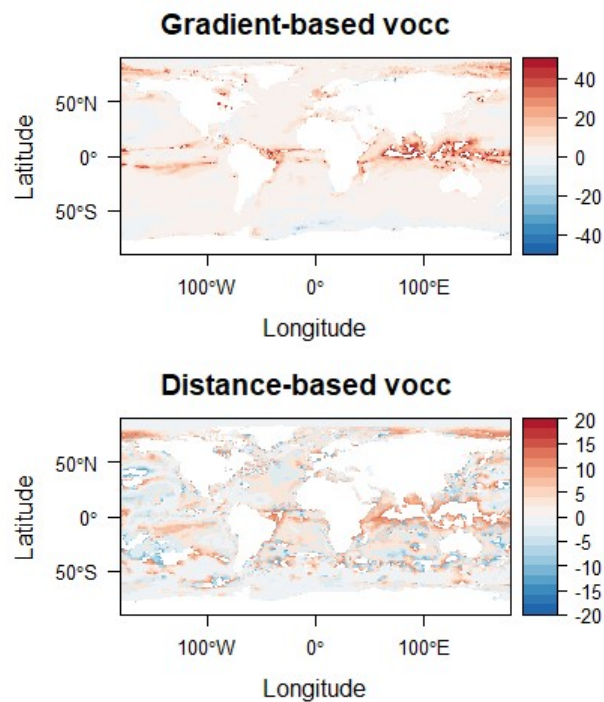
Produce the observed vs predicted scatterplots with regression lines (Fig. 2 in Garcia Molinos et al. 2019).

```

# first compare both velocities
my.at <- seq(-50, 50, by = 5)

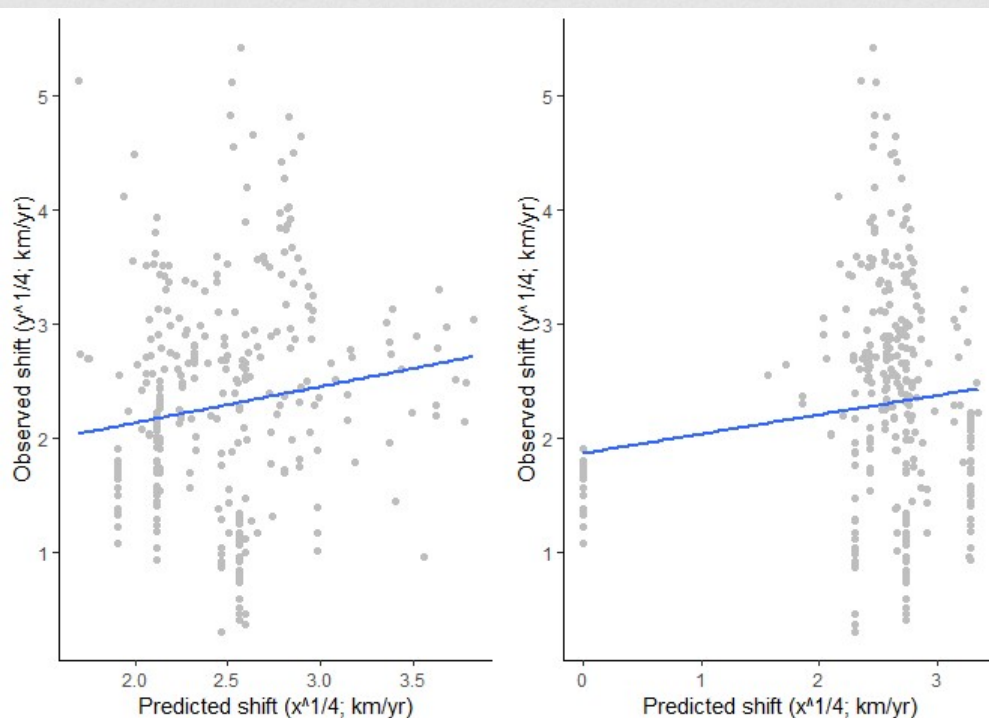
```

```
p1 <- rasterVis::levelplot(gv[[1]], par.settings = BuRdTheme, at=my.at, main = 'Gradient-based
my.at <- seq(-20, 20, by = 2)
p2 <- rasterVis::levelplot(dv, par.settings = BuRdTheme, at=my.at, main = 'Distance-based vocc
gridExtra::grid.arrange(p1, p2, ncol=1)
```



scatter plots with the resulting regression line

```
p1 <- ggplot(na.omit(marshift), aes(x=I((GV*10)^(1/4)), y=Shift^(1/4))) + geom_point(color="grey")
p1 <- theme_classic() + scale_color_brewer(palette="Accent") + labs(x="Predicted shift (x^1/4; km/yr)", y="Observed shift (y^1/4; km/yr)")
p2 <- ggplot(na.omit(marshift), aes(x=I((DV*10)^(1/4)), y=Shift^(1/4))) + geom_point(color="grey")
p2 <- theme_classic() + scale_color_brewer(palette="Accent") + labs(x="Predicted shift (x^1/4; km/yr)", y="Observed shift (y^1/4; km/yr)")
grid.arrange(p1, p2, nrow = 1)
```



Example 2: Analysis of climate exposure and connectivity in the Western Pacific Ocean

In this example we use climate velocity trajectories (based on 1960-2009 mean annual SST) to analyse climate connectivity in the Western Pacific region and calculate the residence time corresponding to the exclusive economic zones in the region as an index of climatic exposure. First, we arrange the raster layers for analysis.

```
# prepare raster layers
vel <- gv[[1]]
ang <- gv[[2]]
mn <- calc(r, mean, na.rm = T)
# generate a velocity layer centered and cropped to study region to extract the initial coordinates
x1 <- crop(gv[[1]], extent(-180, 0, -90, 90))
x2 <- crop(gv[[1]], extent(0, 180, -90, 90))
extent(x1) <- c(180, 360, -90, 90)
velc <- merge(x1, x2)
# crop to the desired extent
# display restricted to +180 Longitude to avoid plotting issues with date line crossing
velc <- crop(velc, c(90, 180, -32, 33))
```

We can now populate the data frame with the cell centroid coordinates for the trajectories and associated input data

```
lonlat <- data.frame(xyFromCell(velc, 1:ncell(velc)))
lonlat$vel <- raster::extract(vel, lonlat)
lonlat$ang <- raster::extract(ang, lonlat[,1:2])
lonlat$mn <- raster::extract(mn, lonlat[,1:2])
lonlat <- na.omit(lonlat)
```

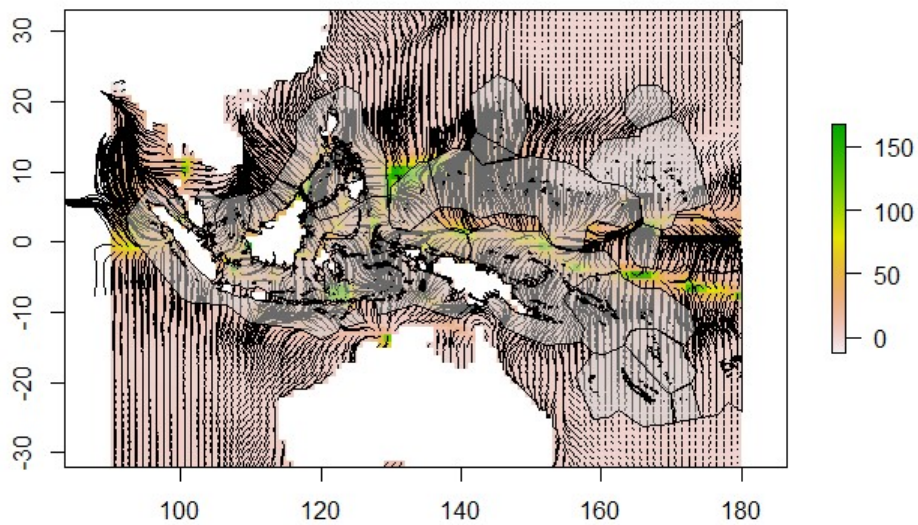
Let's calculate the trajectories with parallel processing to demonstrate how this can be used to speed things up (especially useful when dealing with fine resolutions or large extents).

```
cores = detectCores()
ncores = cores[1]-1
cuts <- cut(1:nrow(lonlat), ncores)
cl <- makeCluster(ncores)
registerDoParallel(cl)
traj <- foreach(x = levels(cuts), .combine = rbind, .packages = c('raster', 'sp', 'rgeos', 'geosplines')) %do{
  stopCluster(cl)
}
```

Plot them over the climate velocities and the EEZ polygons from the EEZs data set (Fig. 3a in Garcia Molinos et al. 2019)

```
# simplify polygons to speed plotting up
eez_simp <- rgeos::gSimplify(EEZs, tol = 0.5, topologyPreserve = TRUE)
plot(velc)
# create the spatial object with the trajectories and plot them together with the EEZ polygons
lns <- trajLine(x = traj)
```

```
plot(lns, add = TRUE)
plot(eez_simp, col = scales::alpha(rgb(211, 211, 211, maxColorValue = 255), 0.5), add=TRUE)
```



We now calculate the trajectory classes and residence times for each EEZ using the traj25 data set.

```
# classify trajectories (16 trajectories starting from each 1-deg cell cell)
clas <- trajClas(traj25, vel, ang, mn, trajSt = 16, tyr = 50, nml = 20, sml = 100, Nend = 45, l
# Extract proportions by categories for each EEZ
v <- data.table(raster::extract(clas[[7]], EEZs, df = TRUE))
v[, TrajClas := as.character(TrajClas)]
v[, ID := as.ordered(ID)]
# proportions by class
d <- prop.table(table(v),1)
# residence times by EEZ
EEZa <- resTime(EEZs, vel, areapg = NA)
```

Finally let's plot the category proportions as pie charts on top of each EEZ, the size of the chart being proportional to their respective residence time (Fig. 3b in Garcia Molinos et al. 2019).

```
D <- data.table(d) # put data in long format
# add EEZ names for reference
D[,name := as.character(EEZs$Territory1)[as.numeric(ID)]]
D[,RT := as.character(EEZa$resTim)[as.numeric(ID)]]
# prepare data frame to plot the pie charts with
dt <- as.data.frame.matrix(d)
dt$country <- as.character(EEZs$Territory1)
dt[,c("x","y")] <- coordinates(EEZs)
dt$RT <- EEZa$resTim
# generate the plot
plot(velc)
plot(eez_simp, add=TRUE)
mycol = c(scales::alpha(rgb(192, 192, 192, maxColorValue = 255), 0.5), scales::alpha(rgb(204, ,
```

```
# mylab = c("Non-moving", "Slow-moving", "Internal Sink", "Boundary sink",
#           "Source", "Internal sink", "Corridor", "Divergence", "Convergence")
for(i in 1:35){add.pie(z = as.numeric(dt[i, 1:9]), x = dt[i,"x"], y = dt[i,"y"], radius = log(
```

