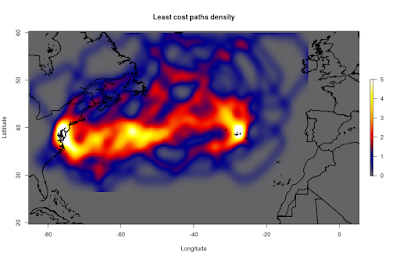
Hi there!

Our manuscript about rWind R package has been recently accepted for publication in [Ecography](https://onlinelibrary.wiley.com/doi/pdf/10.1111/ecog.03730)! As you know, rWind is a tool used to download and manage wind data, with some utilities that make easy to include wind information in ecological or evolutionary analyses ([or others!](https://www.tandfonline.com/doi/abs/10.1080/15230406.2017.1403376)).

Though there are several examples in the Supporting material of the publication, and others in the [vignette](https://cran.r-project.org/web/packages/rWind/vignettes/Shortest_paths.html) included in the CRAN version, here you have a full tutorial with the main utilities of rWind… enjoy!

**Computing wind connectivity of Azores Archipelago**

In this example, we will compute wind connectivity of Azores Islands with different points of Europe, Africa and North America during June 2018. I will divide each task of this tutorial in different sections for clarification.

[](https://i1.wp.com/4.bp.blogspot.com/-r-qzXxqV3WE/W_RCLAkvrtI/AAAAAAAAAS8/GPg3gQtMpjg9flQMBP9uMn85upxC7SbuACLcBGAs/s1600/paths.png?ssl=1)

**Downloading wind time series using lubridate and wind.dl\_2()**

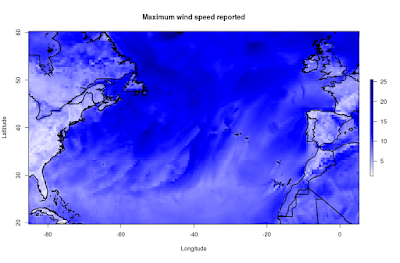
In the last version of rWind (1.0.4), we can use lubridate to create a list of dates which will be used latter with wind.dl\_2 to download wind time series. The object obtained with wind.dl\_2 is an “rWind\_series” “list”. This object can be easily used with other rWind functions as we’ll see later.

# Make sure that you have the last version of rWind   
   
 devtools::install\_github("jabiologo/rWind")   
 library(rWind)   
 library(lubridate)   
   
 # Here, we use ymd\_hms from lubridate package to create a sequence of dates   
   
 dt <- seq(ymd\_hms(paste(2018,6,1,12,00,00, sep="-")),   
 ymd\_hms(paste(2018,6,30,12,00,00, sep="-")),by="1 days")   
   
 # Now we can use wind.dl\_2 with this sequence of dates. Have into account   
 # that it could take a while, they are 30 datasets and it's a big area.   
   
 ww <- wind.dl\_2(dt,-85,5,20,60)

**Obtaining maximum wind speed during the time series with dplyr**

We can use the rWind function called “tidy” to play with the “rWind\_series” “list” object to obtain, for example, the maximum speed reported for each cell in the study area during the time series.

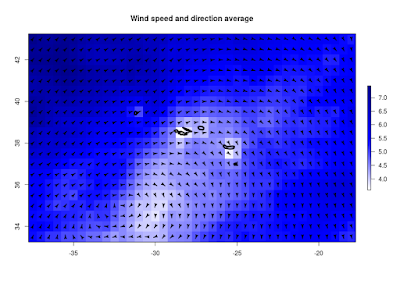
# After tidy our wind data, we can use dplyr utilities to easily   
 # obtain several metrics from our wind data. Notice that wind average   
 # is a special case, and it will be discussed later   
   
 t\_ww <- tidy(ww)   
   
 library(dplyr)   
 g\_ww <- t\_ww %>% group\_by(lat, lon)   
   
 # Now, we select only the maximum speed values for each coordinate   
   
 max\_ww <- g\_ww %>% summarise(speed = max(speed))   
 maxw <- cbind(max\_ww$lon, max\_ww$lat, max\_ww$speed)   
 head(maxw)   
   
 # We can also check the maximum speed value in the whole study area   
   
 max(maxw[,3])   
   
 # Finally, we can transform this data into a raster layer   
   
 rmax <- rasterFromXYZ(maxw)   
 acol <- colorRampPalette(c("white", "blue", "darkblue"))   
 plot(rmax, col=acol(1000), main= "Maximum wind speed reported",   
 xlab="Longitude", ylab="Lattitude")   
   
 library(rworldmap)   
 lines(getMap(resolution = "high"), lwd=2)

[](https://i0.wp.com/4.bp.blogspot.com/-h9C1SQKcfYc/W_RLcm49ydI/AAAAAAAAATQ/nHaIF2PZjugd9JGgfx61aTF_7QTA1VrCQCLcBGAs/s1600/max_speed.png?ssl=1)

**Computing averages of wind speeds and directions**

Since wind data is a vectorial magnitude given in two components (U and V), to compute speed and direction averages we should use both vectors. To make it easier, rWind includes wind.mean function. Notice that wind.mean is specially developed to work with wind.dl\_2 outputs, so it will require “rWind\_series” object as input.

# We can easily use wind.mean() function   
   
 w\_mean <- wind.mean(ww)   
   
 # And then use wind2raster directly to create a raster layer   
   
 r\_mean <- wind2raster(w\_mean)   
  
 # We can plot a subset of this raster around Azores Islands.   
 # Using "arrowDir" we can include arrows for wind direction with   
 # "Arrowhead" function from "shape" R package   
   
 plot(r\_mean$wind.speed, main= "Wind speed and direction average",  
 col=acol(1000), xlim=c(-38,-18), ylim=c(33,43))   
 alpha <- arrowDir(w\_mean)   
 library(shape)   
 Arrowhead(w\_mean$lon, w\_mean$lat, angle=alpha,   
 arr.length = 0.2, arr.type="curved")   
 lines(getMap(resolution = "low"), lwd=4)

[](https://i1.wp.com/2.bp.blogspot.com/-p0EVkGbqvxs/W_ROtMgk2pI/AAAAAAAAATc/5yGpmEJ8nfIvdjbVeRRyLhdF8xH3FiejQCLcBGAs/s1600/azores.png?ssl=1)

**Wind connectivity between Azores and mainland**

To complete this tutorial, we will use the downloaded wind data to compute wind connectivity between several points in mainland and Azores Archipelago. This kind of measures can be useful in order to establish theories about island colonization by plants or other organisms that could be influenced by wind currents. We will also use this connectivity to compute least cost paths between two locations, in order to virtually plot a wind highway connecting mainland with Azores islands.

# First, we define our locations. 7 locations in the East   
 # and 7 in the West. The last one is a point in the Azores islands   
   
 loc <- matrix(c(-7, -4, -1, -9, -6,-10,-15,   
 -60, -56,-63,-76,-76,-81,-80, -27,   
 55, 50, 45, 40, 35, 30, 25,   
 55, 50, 45, 40, 35, 30, 25, 39), 15, 2)   
   
 colnames(loc) <- c("lon", "lat")   
 rownames(loc) <- c(seq(1,14,1), "Azores")   
   
 # Check how "loc" is looking   
   
 tail(loc)   
   
 # You can plot them   
 #plot(r\_mean$wind.speed, col=acol(1000))   
 #lines(getMap(resolution = "low"), lwd=2)   
 #points(loc, col="red4", pch = 19, cex = 2)

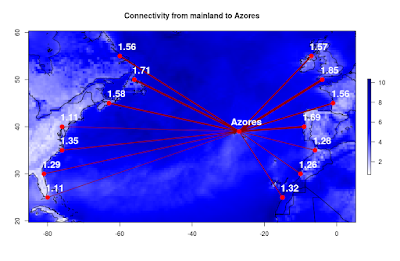
Now, we’ll execute the next tasks:

1. We’ll use wind2raster to obtain raster layers of wind direction and speed of each day in the time series.
2. Then, for each day, we‘ll compute the conductance matrix, a matrix with connectivity values between every single cell of the entire study area.
3. Later, with each conductance matrix, we’ll compute the cost between our locationsand will store them in a cost\_list object.
4. Finally, for each day, if the Cost is not Infinite, we’ll get the shortest path between two selected locations, and we’llstore it into a “paths” object.

# 1) Raster layers from wind time series   
   
 layers <- wind2raster(ww)   
   
 # 2) Conductance from wind direction and speed   
   
 Conductance <- flow.dispersion(layers, type="passive",   
 output="transitionLayer")   
   
 # 3) and 4) Using a loop to compute cost between   
 # locations and least cost paths between location 11   
 # and Azores islands (as an example).   
   
 cost\_list <- array(NA\_real\_, dim=c(15,15,30))   
 paths <- list(1:30)   
   
 library(gdistance)   
   
 for (i in 1:30){   
 cost\_list[,,i] <- costDistance(Conductance[[i]],loc)   
 if (costDistance(Conductance[[i]],loc[11,] , loc[15,]) != Inf){   
 paths[[i]] <- shortestPath(Conductance[[i]], loc[11,], loc[15,],   
 output="SpatialLines")   
 print(i)   
 }   
 }

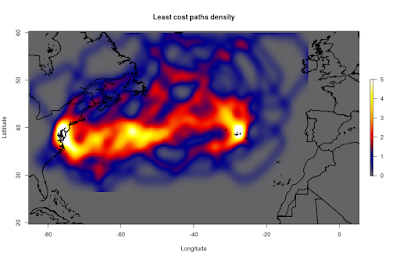
Now, we will manage a little bit the results to plot them in a cool way.

# Transform costs in connectivity   
   
 connectivity <- 1/cost\_list   
   
 # Obtain connectivity averages   
   
 conn\_avg <- apply(connectivity, c(1, 2), mean, na.rm = TRUE)   
   
 # Here we filter only connectivity from mainland to Azores   
 # As connectivity has arbitrary magnitude and units, we will   
 # multiply connectivity values 1000 times to avoid decimal zeros.   
   
 conn\_azores <- conn\_avg[,15]   
 conn\_azores <- conn\_azores \*1000   
   
 # Finally, we can plot connectivity values relative to the width   
 # of the connections between mainland and Azores. You can power   
 # the values to increase the differences between them.   
   
 plot(r\_mean$wind.speed, main= "Connectivity from mainland to Azores",  
 col=acol(1000))   
 lines(getMap(resolution = "low"), lwd=1)   
 for (i in 1:14){   
 lines(rbind(loc[i,], loc[15,]), lwd = conn\_azores[i]^3, col="red4")   
 }   
 points(loc, col="red", pch = 19, cex = 1.5)   
 text( loc+2, labels=c(round(conn\_azores[1:14], 2),"Azores"),   
 col="white", cex= 1.5, font=2)

[](https://i1.wp.com/3.bp.blogspot.com/-CVH8P2JnOW8/W_SS5L5v5eI/AAAAAAAAATw/OcMXT6cKAGQlzw6H0WXmgNNWSbaGwiKzwCLcBGAs/s1600/connectivity.png?ssl=1)

Last but not least, we can use the shortest paths stored for each day to plot a density heatmap. This can help us to visualize wind corridors across our study area (our wind highways!).

# We'll use spatstat library to convert the shortest   
 # paths lines into an psp object.   
   
 library(spatstat)   
   
 paths\_clean <- paths[!sapply(paths, is.null)]   
 paths\_merged <- paths\_clean[[1]]   
 for (h in 2:length(paths\_clean)) {   
 paths\_merged <- rbind(paths\_merged,paths\_clean[[h]])   
 }   
 paths\_psp <- as(paths\_merged, "psp")   
   
 # Now, we can obtain a kernel density of the paths   
 # and convert it into a raster layer that can be plotted.   
 # You can play with sigma value to modify the smoothing.   
   
 lines\_kernel <- density(paths\_psp, sigma=1, dimyx=c(350,410))   
 kernel <- raster(lines\_kernel)   
 #kernel[kernel<(maxValue(kernel)\*0.1)]<-NA   
   
 ext <- extent(r\_mean$wind.speed)   
 kernel <- extend(kernel, ext)   
 kernel[is.na(kernel)] <- minValue(kernel)   
   
 acol <- colorRampPalette(c("grey40","darkblue","red2",  
 "orange","yellow","white"))   
 plot(kernel, col=acol(1000), zlim=c(-0.01,5),   
 main = "Least cost paths density", xlab="Longitude",   
 ylab="Lattitude")   
 lines(getMap(resolution = "high"), lwd=2)

[](https://i2.wp.com/1.bp.blogspot.com/-r-qzXxqV3WE/W_RCLAkvrtI/AAAAAAAAATA/H4p61GsR79Et4hq3KFUhYOs4E_qFaM9vwCPcBGAYYCw/s1600/paths.png?ssl=1)

Isn’t it cool?

 That’s all for now!