Generate spatial passage layer for species distribution modelling

This directory contains code and input data for generating a layer representing mean passage probability among spatially balanced points* across the northwest shelf.

Mean passage probability was estimated based on the ocean current bearing (direction) and ocean current speed which can produce asymmetrical routes between any two points. As such, pairwise mean passage probability values were estimated and then visualised. The final output is a `.csv` file which can be used as input/predictor layer for species distribution modelling. These mean passage probability values are also visualised into accompanying `.pdf` output.

The code was written in R and executed using the University of Adelaide High Performance Computer (Phoenix HPC).

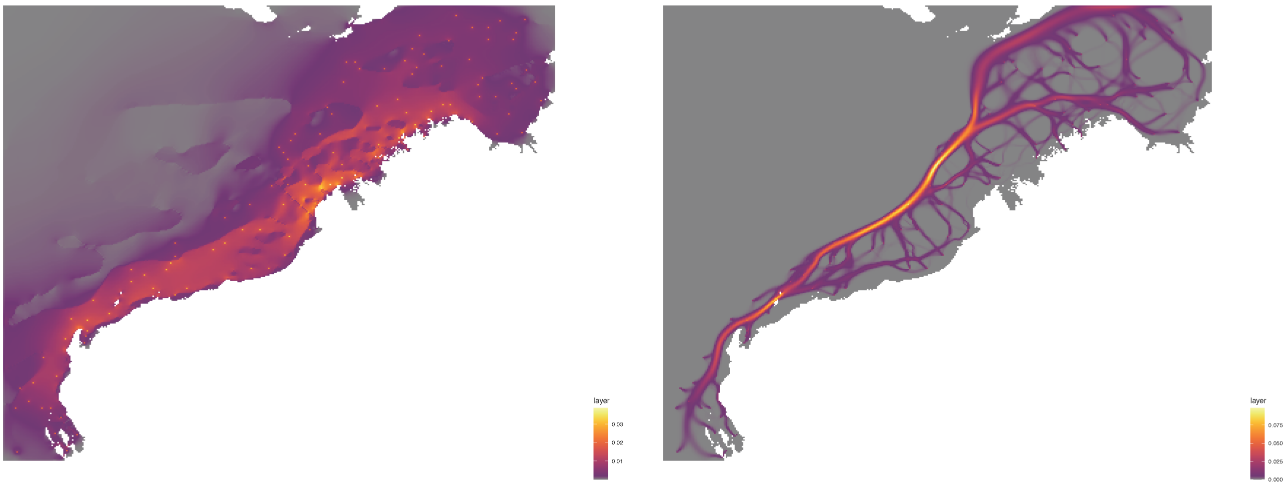
*Spatially balanced points plus manually selected points in Shark Bay and Exmouth Gulf to explicitly include such localities.

Run in Phoenix HPC

The R scripts and associated bash scripts are stored in: `/hpcfs/users/a1235304/atm_passage`. Each takes about 7.5 hours to run.

```
sbatch sbs_bearing_bash.sh
sbatch sbs_speed_bash.sh
```

Here is a preview of the output:



Estimated mean passage probability across the northwest shelf based on ocean current bearing (direction) (left); speed (right).

From here, I go through the R code.

Download/Load ocean current bearing and speed datasets

We will obtain our ocean current bearing and speed datasets from BioOracle via the R package `biooracler`. BioOracle layers are at the spatial resolution of 0.05 x 0.05 decimal degrees and of decadal temporal resolution.

We have code that will download these BioOracle layers each time it is run (see either sbs_bearing_script.R or sbs_speed_script.R). However, we may want to save these layers to file so we can access it immediately when running the job through the HPC.

```
# load required packages
library(gdistance)
```

```
library(tidyverse)
library(ggplot2)
library(raster)
library(viridis)
library(ggthemes)
library(biooracler)
library(sf)
library(terra)

### To circumvent cURL timeout:
### *** The raster files have been downloaded and now exist in rasterfiles/ dir *** ###

# Load the tif file (class will be SpatRaster)
swd_layer_raster <- rast("/hpcfs/users/a1235304/atm_passage/rasterfiles/swd_layer_raster.tif")
#sws_layer_raster <- rast("/hpcfs/users/a1235304/atm_passage/rasterfiles/sws_layer_raster.tif")

# Convert to RasterLayer
swd_layer_raster <- raster(swd_layer_raster)
#sws_layer_raster <- raster(sws_layer_raster)
```

The code block above is from the standalone script for ocean current bearing (direction) and as such lines for `sws_*` have been commented out (and for the rest of the scripts presented below). The standalone script for ocean current speed (sbs_speed_script.R) is basically the same but with speed lines activated.

Generate spatially balanced points (plus manually selected points for Shark Bay and Exmouth Gulf)

We will now place spatially balanced points within the northwest shelf boundary. We are going to use the northwest shelf shapefile obtained from Vinay Udyawer.

```
# load shapefile
nw_shelf <- st_read("/hpcfs/users/a1235304/atm_passage/shapefiles/nw_shelf/NWShelf.shp", quiet = TRUE) %>%
st_transform(4326)

library(sf)
nw_shelf_utm <- st_transform(nw_shelf, crs = 32750)

library(dssduoa)
nw_shelf_region <- make.region(shape = nw_shelf_utm)

library(spsurvey)
seed <- 100
set.seed(seed) # we set seed to track iteration
n_base <- 100
ecoflow_pts <- grts(nw_shelf_utm, n_base = n_base)

plot(ecoflow_pts)

ecoflow_pts_sf <- st_as_sf(ecoflow_pts$sites_base)

ecoflow_pts_lon <- ecoflow_pts$sites_base$lon_WGS84
ecoflow_pts_lat <- ecoflow_pts$sites_base$lat_WGS84
```

```
ecoflow_pts_coords <- cbind(ecoflow_pts_lon,ecoflow_pts_lat)
colnames(ecoflow_pts_coords) <- c("longitude","latitude")
df_ecoflow_pts_coords <- as.data.frame(ecoflow_pts_coords)

# introduce manually selected points in EG and SB
manual_pts <-
  data.frame(longitude = c(114.37139,114.31644,114.29309,113.69914,113.25505,113.45073),
             latitude = c(-21.86184,-22.08711,-22.31837,-26.38297,-25.73358,-25.13991))

# append manually selected points to sbs generated pts
df_ecoflow_pts_coords <- rbind(df_ecoflow_pts_coords, manual_pts)

# check for duplicate points
dim(df_ecoflow_pts_coords) == dim(unique(df_ecoflow_pts_coords)) # no duplicate points

# generate origin/destination combinations across the 100 points
ecoflow_pts_comb <-
  combn(nrow(df_ecoflow_pts_coords),2) %>%
  t() %>%
  as.matrix()
ecoflow_pts_comb
```

Estimate mean passage probability

We will now estimate mean passage probability among pairwise points. Pairwise points were limited within the northwest shelf boundary but mean passage probability estimates were made throughout the extent of the BioOracle ocean current bearing layer.

```
# bearing
swd_layer_tr <- transition(swd_layer_raster, transitionFunction = mean, directions = 8) %>%
  geoCorrection(type = "c", multpl = F)

ecoflow_bearing_passages <- list()

system.time( # Keep track of how long this takes
  for (i in 1:nrow(ecoflow_pts_comb)) {
    locations <- SpatialPoints(rbind(df_ecoflow_pts_coords[ecoflow_pts_comb[i,1],1:2],   # create origin points
                                     df_ecoflow_pts_coords[ecoflow_pts_comb[i,2],1:2]),  # create destination
(or goal) points, to traverse
                               proj4string = CRS("+init=epsg:4326"))
    ecoflow_bearing_passages[[i]] <- passage(swd_layer_tr,        # run the passage function
                                             origin=locations[1], # set orgin point
                                             goal=locations[2],   # set goal point
                                             theta = 0.00001)     # set theta (tuning parameter, see notes
below)
    print(paste((i/nrow(ecoflow_pts_comb))*100, "% complete"))
  }
)

ecoflow_bearing_passages <- stack(ecoflow_bearing_passages) # create a raster stack of all the passage
probabilities
ecoflow_bearing_passages_overlay <- sum(ecoflow_bearing_passages)/nrow(ecoflow_pts_comb) # calculate average
```

From Mapping Ecological Flow in R: "*In our passage function, we set theta, (θ), a tuning parameter. Extremely low values result in a random walk (equivilant to Circuit Theory), but as θ¸ increases, the passage converges on least cost path. I supplied a value somewhere in the middle.*"

We then supply the code to write our output. We produce a `.csv` file which can be used as input for species distribution modelling and a `.pdf` file for visualisation.

```
### *** save output *** ###

dir <-
paste0("/hpcfs/users/a1235304/atm_passage/output/sbs_bearing_seed",seed,"_",n_base,"pts_",format(Sys.time(),"%Hh
%Mm%Ss"),"/")
dir.create(dir, recursive = TRUE)

# as csv
library(terra)
ovr <- rast(ecoflow_bearing_passages_overlay)
df_ovr <- as.data.frame(ovr, xy = TRUE)

library(utils)
write.csv(df_ovr, file = paste0(dir,
"sbs_bearing_seed",seed,"_",n_base,"pts_",format(Sys.time(),"%Hh%Mm%Ss"),".csv"))

# as pdf

pdf(file = paste0(dir, "sbs_bearing_seed",seed,"_",n_base,"pts_",format(Sys.time(),"%Hh%Mm%Ss"),".pdf"), height
= 8.5, width = 11)

colors <- c("grey50", viridis_pal(option="inferno", begin = 0.3, end = 1)(20))
ggplot(as.data.frame(ecoflow_bearing_passages_overlay, xy=T)) +
  geom_raster(aes(x=x,y=y,fill=layer)) +
  scale_fill_gradientn(colors = colors, na.value = NA) +
  #geom_point(data=as.data.frame(samp_coords), aes(x=longitude, y=latitude), size=1, col="red") +
  theme_map() +
  theme(legend.position = "right")
```