

Archival tags

James T. Thorson

```
library(walk)

# Additional packages for demo below
library(terra)
library(sf)
library(rnaturalearth)
library(ggplot2)
library(Matrix)
library(ggplot2)
library(tidyterra)
library(ggspatial)
library(lubridate)
library(animation)
```

walk estimates a preference function and variation in diffusion rates using a “data likelihood” that represents locational error from tagging or archival tags. We demonstrate this using data for a tagged Pacific cod in the Aleutian Islands.

```
data(pcod_archival_tag)

# Extract pieces
bathy_terra <- rast(pcod_archival_tag$bathymetry)
likelihood_terra <- rast(pcod_archival_tag$data_likelihood)

# Get land layer
ak <- ne_states( "united states of america", return="sf")
ak <- ak[pmatch("Alas", ak$name_en),]
ak <- st_transform( ak, crs=st_crs("+proj=aea +lat_0=50 +lon_0=-154 +lat_1=55 +lat_2=65 +x_0=0 +y_0=0 +"))
ak <- st_geometry(ak)

# change resolution
fact <- 2
bathy_terra <- terra::aggregate( bathy_terra, fact = fact )
likelihood_terra <- as.array( terra::aggregate(likelihood_terra, fact=fact) )

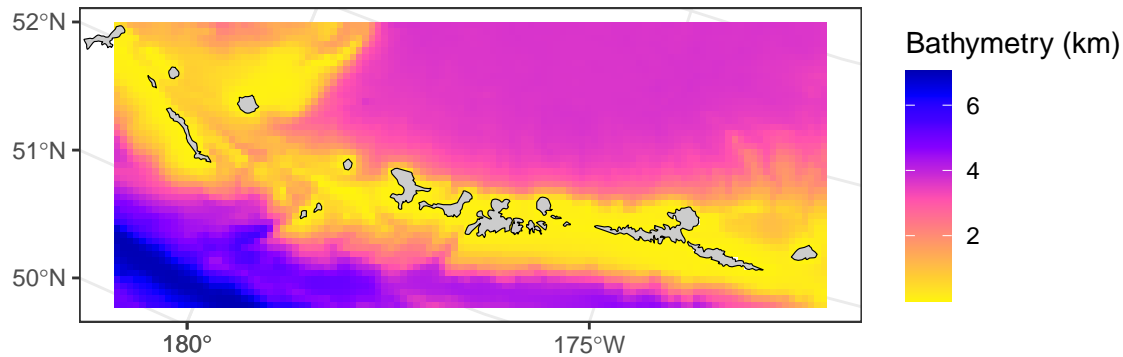
# Eliminate land cells
bathy_terra[bathy_terra<=1] <- NA
bathy_terra$bathy_km <- bathy_terra/1000
bathy_terra$bathy_km2 <- bathy_terra$bathy_km^2
bathy_terra$bathy_t <- terra::ifel(bathy_terra$bathy_km>3, 3, bathy_terra$bathy_km)

# Plot
ggplot() +
  geom_spatraster(data=bathy_terra$bathy_km) +
```

```

scale_fill_gradientn(colors=rev(sf.colors(10)), na.value = "transparent", name="Bathymetry (km)" ) +
# scale_fill_distiller(palette="YlOrRd", direction = 1, na.value = "transparent", name="UD") +
annotation_spatial(ak, fill=gray(0.8), color=1) +
scale_y_continuous(breaks=seq(-180,180,1)) +
scale_x_continuous(breaks=seq(-180,180,5)) +
theme_bw()

```



Next we create data inputs in the format expected:

```

L <- t(apply(likelihood_terra, MARGIN=3, FUN=function(mat){as.vector(t(mat))}))
L <- Matrix(L) |> as("CsparseMatrix")
pdata <- list(L=L)
pdata$times <- data.frame(
  obs=1:nrow(L),
  timestamp = seq(mdy_hms("2/21/2019 12:00:00"), mdy_hms("5/23/2019 12:00:00"), "1 day")
)
pdata$times$dt = c(0, diff(pdata$times$timestamp))

```

```

units(pdata$times$dt) <- "days"
attr(pdata$times, "time_unit") <- "days"

# Format habitat for {walk}
walk_data <- make_walk_data(pdata, bathy_terra, grad=c("bathy_km", "bathy_km2"), rast_mask = bathy_terra

### Bathymetry differences for preference function.
walk_data$q_m$d_bathy <- (walk_data$q_m$bathy_km - walk_data$q_m$from_bathy_km) / (res(bathy_terra)[1]/100)
walk_data$q_m$d_bathy2 <- (walk_data$q_m$bathy_km^2 - walk_data$q_m$from_bathy_km^2) / (res(bathy_terra)[1]/100)

```

Then we can fit the continuous-time Markov chain as a Hidden Markov model (HMM):

```

# Diffusion depends on average d_bathy more like traditional CTMC
fit <- fit_ctmc(walk_data,
  model_parameters = list(
    q_r = list(form=~ bathy_km), # residency model
    q_m = list(form=~ d_bathy + d_bathy2) # preference model
  ),
  reals = FALSE, fit=TRUE, control=list(trace=1)
)

#> parchanged = FALSE
#> Parameter scaling:[1] 1 1 1 1
#> 0: 468.61164: 0.00000 0.00000 0.00000 0.00000
#> 1: 359.90560: 1.93765 0.752372 -0.0209094 -0.0302032
#> 2: 330.44935: 0.826297 5.70749 -1.30614 -1.98650
#> 3: 322.59199: 0.708764 7.68688 -1.40973 -1.78057
#> 4: 312.97248: 0.601590 11.6437 -1.47184 -1.07525
#> 5: 309.22073: 0.595233 14.8681 -1.94882 -1.68088
#> 6: 307.34100: 0.595474 18.0754 -1.78259 -1.00751
#> 7: 306.91596: 0.650976 19.6832 -1.83549 -1.24209
#> 8: 306.81719: 0.709134 20.6545 -1.72420 -1.28039
#> 9: 306.78673: 0.767307 20.9478 -1.58402 -1.35745
#> 10: 306.75299: 0.868864 21.0346 -1.30828 -1.51000
#> 11: 306.73323: 0.947135 20.8435 -1.07048 -1.64400
#> 12: 306.72398: 0.992208 20.5618 -0.906839 -1.73871
#> 13: 306.72179: 1.00107 20.3688 -0.837176 -1.77891
#> 14: 306.72138: 0.993544 20.3399 -0.826707 -1.78235
#> 15: 306.72066: 0.973600 20.3331 -0.795037 -1.79099
#> 16: 306.72026: 0.963136 20.3545 -0.765784 -1.80090
#> 17: 306.71961: 0.942774 20.4398 -0.679224 -1.83360
#> 18: 306.71953: 0.943035 20.4690 -0.658334 -1.84339
#> 19: 306.71952: 0.945543 20.4774 -0.655103 -1.84589
#> 20: 306.71952: 0.946012 20.4770 -0.655949 -1.84569
#> 21: 306.71952: 0.946050 20.4768 -0.656080 -1.84565

```

After model fitting, we can then extract the utilization distribution:

```

# Get UD
ud <- get_lim_ud(fit)
# Get parameter estimates
beta_q_m <- fit$results$beta$q_m$est
# Calculate preference function from parameters
ud$pref <- beta_q_m[1]*walk_data$q_r$bathy_km + beta_q_m[2]*walk_data$q_r$bathy_km^2

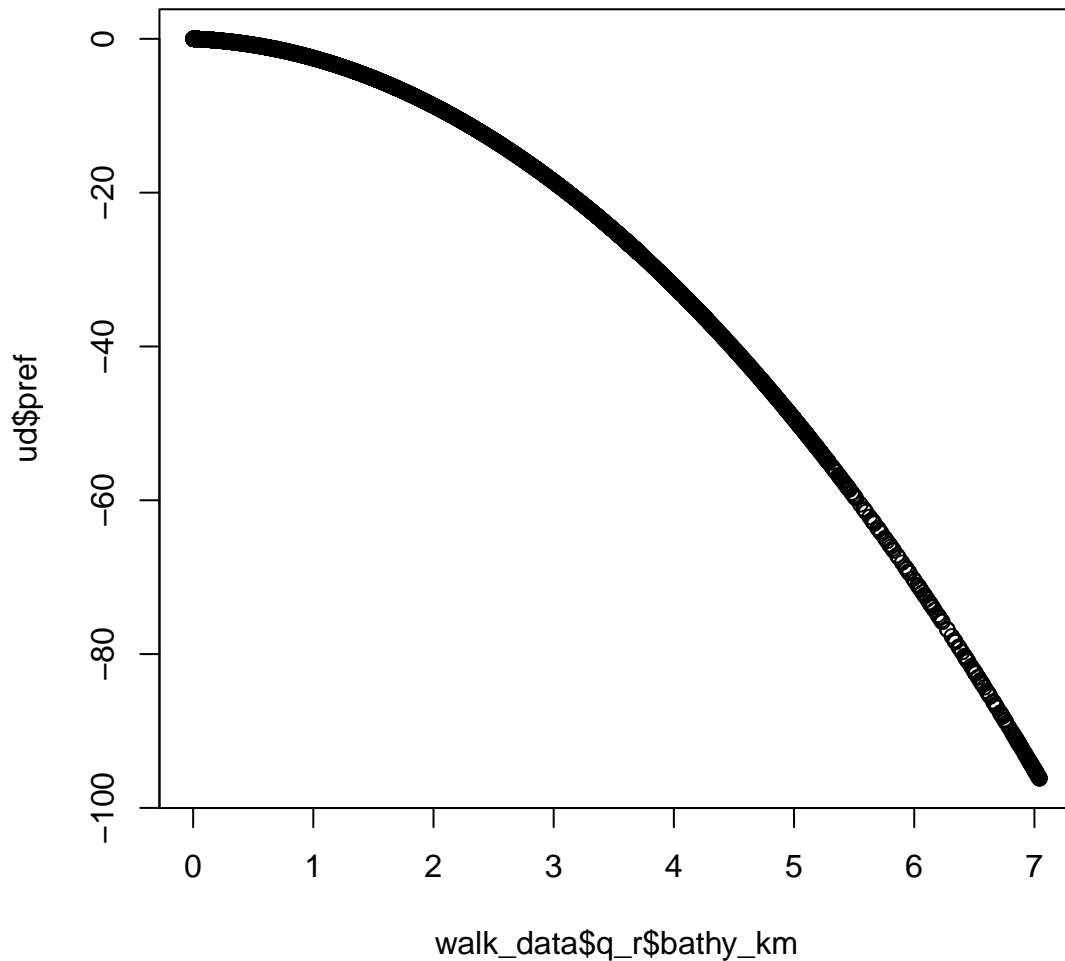
```

```

# Get movement-rate (generator) matrix
Q <- get_Q(fit)
# Get expected residence time from diagonal of movement-rate matrix
ud$res <- -1/diag(Q)

# Plot preference as function of bathymetry
plot(walk_data$q_r$bathy_km, ud$pref)

```



Finally, we can plot the

```

# Create raster of output
cells <- rast(bathy_terra)
cells[['ud']] <- 0
cells[ud$cell] <- ud$ud
cells[['pref']] <- 0
cells[['pref']][ud$cell] <- ud$pref
cells[['res']] <- 0

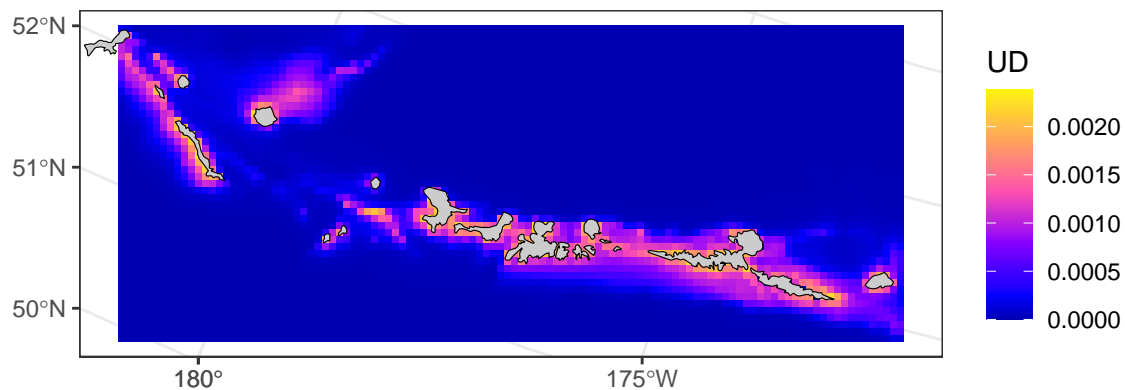
```

```

cells[['res']][ud$cell] <- ud$res

# Plot utilization distribution
ggplot() +
  geom_spatraster(data=cells$ud) +
  scale_fill_gradientn(colors=sf.colors(10), na.value = "transparent", name="UD") +
  # scale_fill_distiller(palette="YlOrRd", direction = 1, na.value = "transparent", name="UD") +
  annotation_spatial(ak, fill=gray(0.8), color=1) +
  scale_y_continuous(breaks=seq(-180,180,1)) +
  scale_x_continuous(breaks=seq(-180,180,5)) +
  theme_bw()

```



```

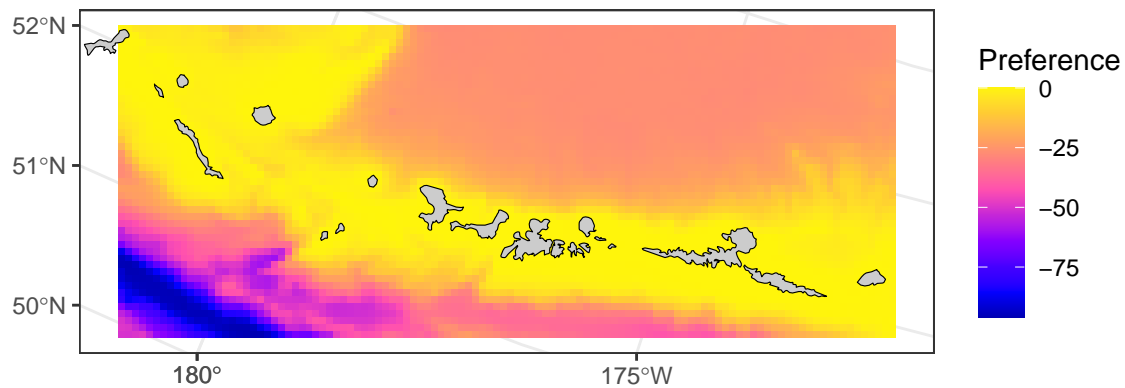
# Plot preference function
ggplot() +
  geom_spatraster(data=cells$pref) +
  scale_fill_gradientn(colors=sf.colors(10), na.value = "transparent", name="Preference") +

```

```

annotation_spatial(ak, fill=gray(0.8), color=1) +
scale_y_continuous(breaks=seq(-180,180,1)) +
scale_x_continuous(breaks=seq(-180,180,5)) +
theme_bw()

```



```

# Plot residence time
ggplot() +
  geom_spatraster(data=cells$res) +
  scale_fill_gradientn(colors=sf.colors(10), na.value = "transparent", name="E[residence] (d)") +
  annotation_spatial(ak, fill=gray(0.8), color=1) +
  scale_y_continuous(breaks=seq(-180,180,1)) +
  scale_x_continuous(breaks=seq(-180,180,5)) +
  theme_bw()

```

