2_phase2_segmentation.R

rfowler

Thu Mar 29 13:55:22 2018

```
#### Required ####
library(maptools)
```

```
## Loading required package: sp
```

```
## Checking rgeos availability: TRUE
```

```
library(rgeos)
```

```
## rgeos version: 0.3-26, (SVN revision 560)
##  GEOS runtime version: 3.6.1-CAPI-1.10.1 r0
##  Linking to sp version: 1.2-5
##  Polygon checking: TRUE
```

```
library(geosphere)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:rgeos':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
library(tidyr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
set.seed(42)

#### Main Function ####
segmentCTS = function(observations,
                      tracks,
                      transects,
                      seg.length = 4,
                      seg.tol = 0.5,
                      seg.min = seg.length * seg.tol,
```

```
                    maxDist = NA) {
#### seg ####
seg = tracks %>%
  distinct(long, lat, piece, transect_id, .keep_all = TRUE) %>%
  group_by(transect_id, piece) %>%
  mutate(
    long_i = lag(long, default = first(long),
                 order_by = order),
    lat_i  = lag(lat, default = first(lat),
                 order_by = order)
  ) %>%

  rowwise %>%
    mutate(dist = distVincentySphere(c(long_i, lat_i), c(long, lat)) / 1000) %>%
    select(-c(long_i, lat_i, order)) %>%
  ungroup %>%

  group_by(transect_id, piece) %>%
  mutate(dist_cuml = cumsum(dist),
         dist_total = max(dist_cuml)) %>%
  select(-dist) %>%

  mutate(
    nseg = ifelse(
      dist_total <= seg.length,
      1,
      ifelse(
        dist_total / seg.length - floor(dist_total / seg.length) >= seg.tol,
        floor(dist_total / seg.length) + 1,
        floor(dist_total / seg.length)
      )
    ),
    dist_extra = dist_total - seg.length * floor(dist_total / seg.length),
    dist_odd = ifelse(
      nseg == 1,
      0,
      ifelse(
        dist_extra < seg.length * seg.tol,
        dist_extra + seg.length,
        dist_extra
      )
    ),
    seg_odd =
      ifelse(dist_odd == 0,
             0,
             ceiling(runif(1, 0, nseg))),
    seg_num = ifelse(
      nseg == 1 | dist_cuml == 0,
      1,
      ifelse(
        dist_cuml <= seg.length * (seg_odd - 1),
        ceiling(dist_cuml / seg.length),
        ifelse(
```

```
          dist_cuml > seg.length * (seg_odd - 1) + dist_odd,
          ceiling(1 + round((
            dist_cuml - dist_odd
          ) / seg.length, 10)),
          seg_odd
        )
      )
    ),
    tot_empty = as.integer(nseg - n_distinct(seg_num))
  ) %>%
  select(-dist_extra)

#### seg.empty ####
seg.empty = seg %>%
  ungroup %>%
  select(piece,
         dataset_id,
         transect_id,
         dist_total,
         nseg,
         dist_odd,
         seg_odd,
         tot_empty) %>%
  distinct %>%
  filter(tot_empty > 0) %>%
  slice(rep(row_number(), tot_empty)) %>%
  select(-tot_empty) %>%
  mutate(empty_seg = 1)

#### seg.all ####
seg.all = seg
seg.all = seg.all %>%
  select(-tot_empty)
seg.all = seg.all %>%
  bind_rows(., seg.empty)
seg.all = seg.all %>%
  group_by(transect_id, piece)
seg.all = seg.all %>%
  mutate(
    seg_num = replace(seg_num, is.na(seg_num), setdiff(1:first(nseg), seg_num)),
    seg_dist = ifelse(
      nseg == 1,
      dist_total,
      ifelse(seg_num == seg_odd, dist_odd, seg.length)
    ),
    seg_dist_cuml = ifelse(
      nseg == 1,
      seg_dist,
      ifelse(
        seg_num >= seg_odd,
        seg.length * (seg_num - 1) + dist_odd,
        seg.length * seg_num
      )
```

```
    )
  )
seg.all = seg.all %>%
  select(-c(dist_total, dist_odd, seg_odd)) %>%
  ungroup %>%
  arrange(dataset_id, transect_id, piece, seg_num, dist_cuml) %>%
  group_by(transect_id, piece) %>%
  mutate(dist_cuml = na.locf(dist_cuml)) %>%
  group_by(transect_id, piece, seg_num) %>%
  mutate(seg_brk = as.integer(ifelse(row_number() == n() & seg_num != nseg,
                                     1,
                                     0))) %>%
  select(-nseg) %>%
  group_by(transect_id, piece) %>%
  mutate(
    long = na.locf(long),
    lat = na.locf(lat),
    long_lead = na.locf(lead(long), na.rm = FALSE, fromLast = TRUE),
    lat_lead = na.locf(lead(lat), na.rm = FALSE, fromLast = TRUE)
  ) %>%
  rowwise %>%
  mutate(
    heading =
      as.numeric(
        ifelse(
          seg_brk == 0,
          NA,
          bearing(
            c(long, lat),
            c(long_lead,
            lat_lead
          ),
        f = 0 )))
  ) %>%
  select(-c(seg_brk, long_lead, lat_lead)) %>%
  ungroup
seg.all = seg.all %>%
  group_by(transect_id, piece, dist_cuml) %>%
  mutate(
    heading = last(heading)) %>%
  group_by(transect_id, piece, seg_num) %>%
  mutate(
    dist_shy =
      as.numeric(ifelse(
        is.na(heading),
        NA,
        seg_dist_cuml - dist_cuml))
  ) %>%
  rowwise %>%
  mutate(coords_end = ifelse(is.na(heading), list(NA), list(
    destPoint(c(long, lat), heading, dist_shy * 1000, f = 0)
  ))) %>%
  select(-c(heading, dist_shy)) %>%
```

```
  ungroup

#### end.pts ####
end.pts = seg.all %>%
  select(-empty_seg) %>%
  filter(!is.na(coords_end)) %>%
  mutate(
    long = unlist(lapply(coords_end, `[[`, 1)),
    lat = unlist(lapply(coords_end, `[[`, 2)),
    dist_cuml = seg_dist_cuml
  ) %>%
  select(-c(coords_end, seg_dist_cuml))

#### seg.ends ####
seg.ends = end.pts %>%
  select(-seg_dist) %>%
  mutate(seg_num = seg_num + 1) %>%
  bind_rows(end.pts, .)

#### seg.all.new ####
seg.all.new = seg.all %>%
  filter(is.na(empty_seg)) %>%
  select(-c(empty_seg, seg_dist_cuml, coords_end)) %>%
  bind_rows(., seg.ends) %>%
  arrange(dataset_id, transect_id, piece, seg_num, dist_cuml) %>%
  select(-dist_cuml) %>%
  mutate(piece =as.integer(piece)) %>%
  group_by(transect_id, piece, seg_num) %>%
  mutate(
    seg_dist = round(max(seg_dist, na.rm = TRUE), 3),
    id = paste(
      sprintf("%02d", transect_id),
      sprintf("%06d", piece),
      sprintf("%02d", seg_num),
      sep = "-"
    )
  ) %>%
  ungroup %>%
  select(-piece) %>%
  as.data.frame() %>%
  filter(seg_dist >= seg.min, seg_dist > 0)

#### Function listLines ####
listLines = function(df) {
  df %>%
    select(long, lat) %>%
    as.data.frame %>%
    Line  %>%
    list
}

#### create linelist ####
linelist = seg.all.new %>%
  group_by(transect_id, id) %>%
```

```
  do(coords = listLines(.))


#### define projHOOM ####
projHOM = "+proj=omerc +lonc=-75 +lat_0=35 +alpha=40 +k_0=0.9996 +ellps=GRS80 +datum=NAD83"



df.linelst = as.data.frame(select(linelist, transect_id))
#### define lineframe ####
lineframe =
  mapply(x = linelist$coords,
         ids = linelist$id,
         function(x, ids) Lines(x, ids)) %>%
  SpatialLines(proj4string = CRS("+proj=longlat")) %>%
  SpatialLinesDataFrame(df.linelst, match.ID = FALSE) %>% ####linelst????
  spTransform(CRS(projHOM))


#### midpoints ####
midpoints = SpatialLinesMidPoints(lineframe) %>%
  spTransform(CRS("+proj=longlat"))
midpoints = midpoints %>%
  as.data.frame %>%
  select(coords.x1, coords.x2) %>%
  rename(mid_long = coords.x1, mid_lat = coords.x2) %>%
  mutate(
    id = sapply(slot(lineframe, "lines"),
                function(x) slot(x, "ID"))
  )

#### segmentation with midpoints ####
seg.mids = seg.all.new %>%
  select(-c(long, lat)) %>%
  distinct %>%
  group_by(transect_id) %>%
  mutate(seg_num = seq.int(n())) %>%
  ungroup %>%
  left_join(., midpoints, by = "id") %>%
  select(-id) %>%
  left_join(
    .,
    select(
      transects,
      transect_id,
      start_dt,
      transect_width_nb,
      survey_type_cd,
      survey_method_cd
    ),
    by = "transect_id"
  )
```

```
#### Function send points to Line ####
assignPointsToLines = function(points, lines, maxDist = NA) {
  if (!is.na(maxDist)) {
    w = gWithinDistance(points, lines, dist = maxDist, byid = TRUE)
    validPoints = apply(w, 2, any)
    points = points[validPoints, ]
  }
  d = gDistance(points, lines, byid = TRUE) # distance matrix of each point to each segment
  seg_num = apply(d, 2, which.min) # position of each nearest segment in lines object
  cbind(points@data, seg_num)
}

#### Function observation to closest line ####
obs2Lines = function(df, lineframe) {
  points = df %>%
    as.data.frame # %>%
    #filter(!is.na(lat) & !is.na(long))
  # apply HOM projection
  coordinates(points) = c("long", "lat")
  proj4string(points) = CRS("+proj=longlat")
  points = spTransform(points, CRS(projHOM))
  lines = lineframe[lineframe@data$transect_id == df$transect_id[1], ]
  assignPointsToLines(points, lines, maxDist)
}

#### Place Observations ####
seg.obs = observations %>%
  filter(transect_id %in% seg.mids$transect_id) %>%
  group_by(transect_id) %>%
  do(obs2Lines(., lineframe)) %>%
  ungroup

#### create segmented df ####
segmented = full_join(seg.mids, seg.obs, by = c("transect_id", "seg_num")) %>%
  mutate(spp_cd = replace(spp_cd, is.na(spp_cd), "NONE")) %>%
  group_by(
    dataset_id,
    transect_id,
    seg_num,
    start_dt,
    seg_dist,
    transect_width_nb,
    mid_long,
    mid_lat,
    survey_type_cd,
    survey_method_cd,
    spp_cd
  ) %>%
  summarise(count = sum(count)) %>%
  spread(spp_cd, count, fill = 0) %>%
  select(everything(), -matches("NONE")) %>%
  ungroup %>%
  mutate(transect_id = as.integer(transect_id))
```

```
  segmented = transects %>%
    select(transect_id, source_dataset_id, segmented_transect_id) %>%
    distinct %>%
    left_join(segmented, ., by = "transect_id") %>%
    select(source_dataset_id,
           segmented_transect_id,
           everything(),
           -dataset_id) %>%
    arrange(transect_id, seg_num) # %>%
    #mutate(transect_id = factor(transect_id))
}
```