Selecting locations for NE Greenlip Timed-swim surveys

Jaime McAllister and Craig Mundy

2023-05-18

Contents

Read in hex layer

Read in grid wide spatial layer, and transform to GDA2020. This hex cell layer contains the majority of fishing activity in the closed blocks.

```
## Read in greenlip abalone wide hex layer

ab.hex <- readRDS(sprintf("C:/Users/%s/University of Tasmania/IMAS-DiveFisheries - Assessments - Documen
        Sys.info()[["user"]]))


## convert to sf
sf.ab.hex <- st_as_sf(ab.hex)

## Transform from GDA94 to GDA2020
sf.ab.hex <- st_transform(sf.ab.hex, GDA2020)
```

Add quartile to dataframe

THe input variable is the total catch in Kg for each cell from 2012 - 2019. We use the dplyr ntile() function to calculate five quantiles (0-20, 20-40, etc). based on total catch.

```
# Filter hex layer to required block and add quartile
ts.hex <- sf.ab.hex %>%
  mutate(subblockno = gsub(" ", "", subblockno)) %>%
  filter(subblockno %in%  c('31A', '31B', '39A','39B')) %>%
  within({
    cell.ntile <- ntile(GLhexcatch, 5)
  })
```

```
outname.ts.hex <- sprintf("C:/Users/%s/University of Tasmania/IMAS-DiveFisheries - Assessments - Documen
st_write(ts.hex, dsn = outname.ts.hex, layer = "pointqt", driver = "GPKG", append = F)
```

```
Deleting layer `pointqt' using driver `GPKG'
Writing layer `pointqt' to data source
  `C:/Users/jaimem/University of Tasmania/IMAS-DiveFisheries - Assessments - Documents/Assessments/GIS/S
Writing 2537 features with 46 fields and geometry type Polygon.
```

```
hex.cell.summary <- ts.hex %>%
  st_set_geometry(NULL) %>%
  group_by(cell.ntile) %>%
  summarise(
    mncatch = mean(GLhexcatch, na.rm = T),
    catch = sum(GLhexcatch, na.rm = T),
    n = n()
  ) %>%
  print()
```

```
# A tibble: 5 x 4
  cell.ntile mncatch   catch     n
       <int>   <dbl>   <dbl> <int>
1          1   0.652    331.   508
2          2   4.78    2429.   508
3          3  19.1     9680.   507
4          4  65.3    33107.   507
5          5 300.    152049.   507
```

randomly sample cells within strata.

We want to randomly select 150 hex cells across the top three quantile strata. The two lower quantile strata contribute 0.0139696 of a total catch of 1.9759551×10^5. The top three quantile groups contribute 0.0489877, 0.1675485 and 0.7694941 of the catch, respectively.

```
set.seed(123)

# set sample size required (i.e. n = 150)
samp.size <- 150

# Exclude oid where a gps was left on while transiting on a motherboat.

oid_exc <- c(1062044, 1062045, 1062604, 1062605, 1062606, 1062607, 1062608, 1062609, 1062610, 1062611, 1

# randomly select sites (NOTE: manually change dataframe name for each block)
cellqt_NE_GL <- ts.hex %>%
  filter(!oid %in% oid_exc &
           cell.ntile > 2) %>%
  group_by(blockno, cell.ntile)  %>%
  st_as_sf() %>%
  subsample.distance(size = samp.size, d = 50) %>%
  st_as_sf() %>%
  ungroup()

  # quick summary of sites x block x strata
table(cellqt_NE_GL$blockno, cellqt_NE_GL$cell.ntile)
```

```
    3  4  5
31 24 37 19
39 19 27 24
```

Examine Nearest Neighbour distances

We find the k=1 Nearest Neighbour and then calculate the distance between a cell and its closest neighbor. Centroids of hex cells are ~ 107m apart.

```
knear <- knearneigh(st_centroid(cellqt_NE_GL), k = 1, longlat = FALSE, use_kd_tree = TRUE)
neighbors <- knn2nb(knear)

summary.nb(neighbors, st_coordinates(st_centroid(cellqt_NE_GL)), longlat = NULL, zero.policy=TRUE)
```

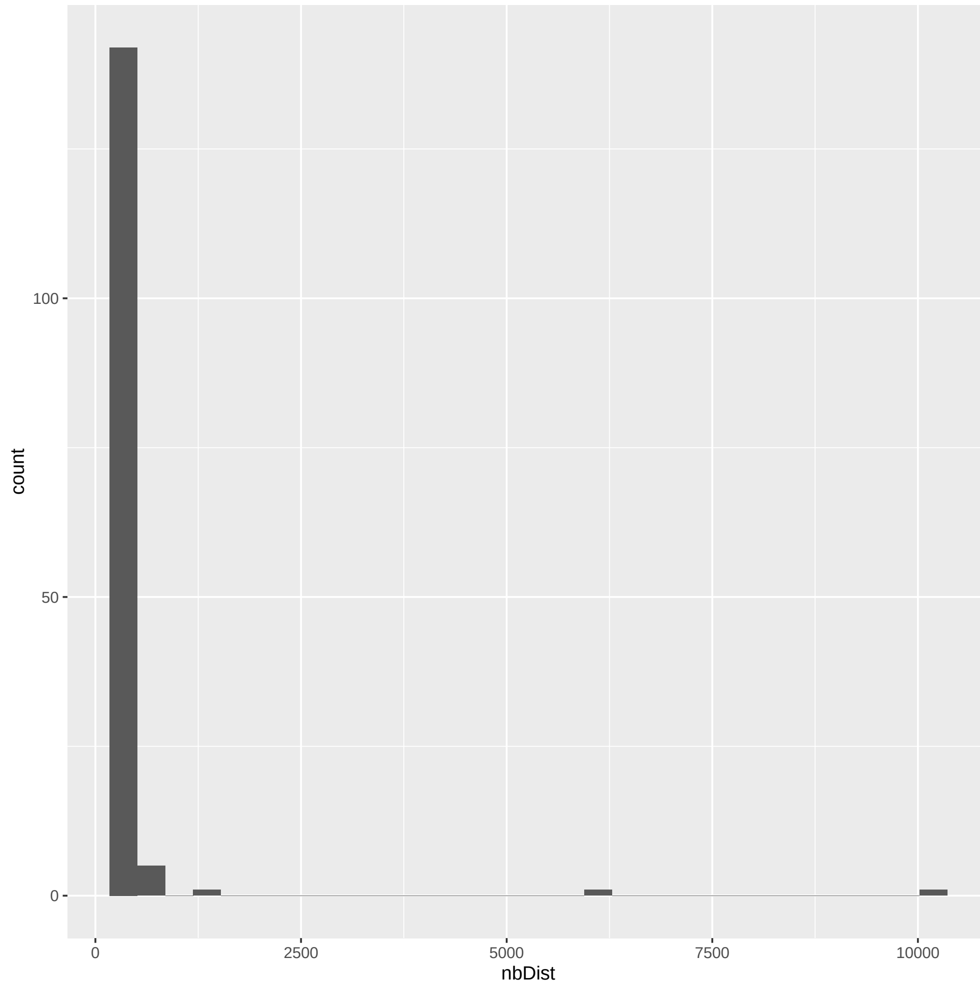
```
Neighbour list object:
Number of regions: 150
Number of nonzero links: 150
Percentage nonzero weights: 0.6666667
Average number of links: 1
Non-symmetric neighbours list
Link number distribution:

  1
150
150 least connected regions:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
150 most connected regions:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
Summary of link distances:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  186.1   186.1   214.9   365.5   284.3 10034.1

  The decimal point is 3 digit(s) to the right of the |

   0 | 222222222222222222222222222222222222222222222222222222222222222222222+60
   0 | 5556667
   1 | 3
   1 |
   2 |
   2 |
   3 |
   3 |
   4 |
   4 |
   5 |
   5 |
   6 | 2
   6 |
   7 |
   7 |
   8 |
   8 |
   9 |
```

```
   9 |
  10 | 0
```

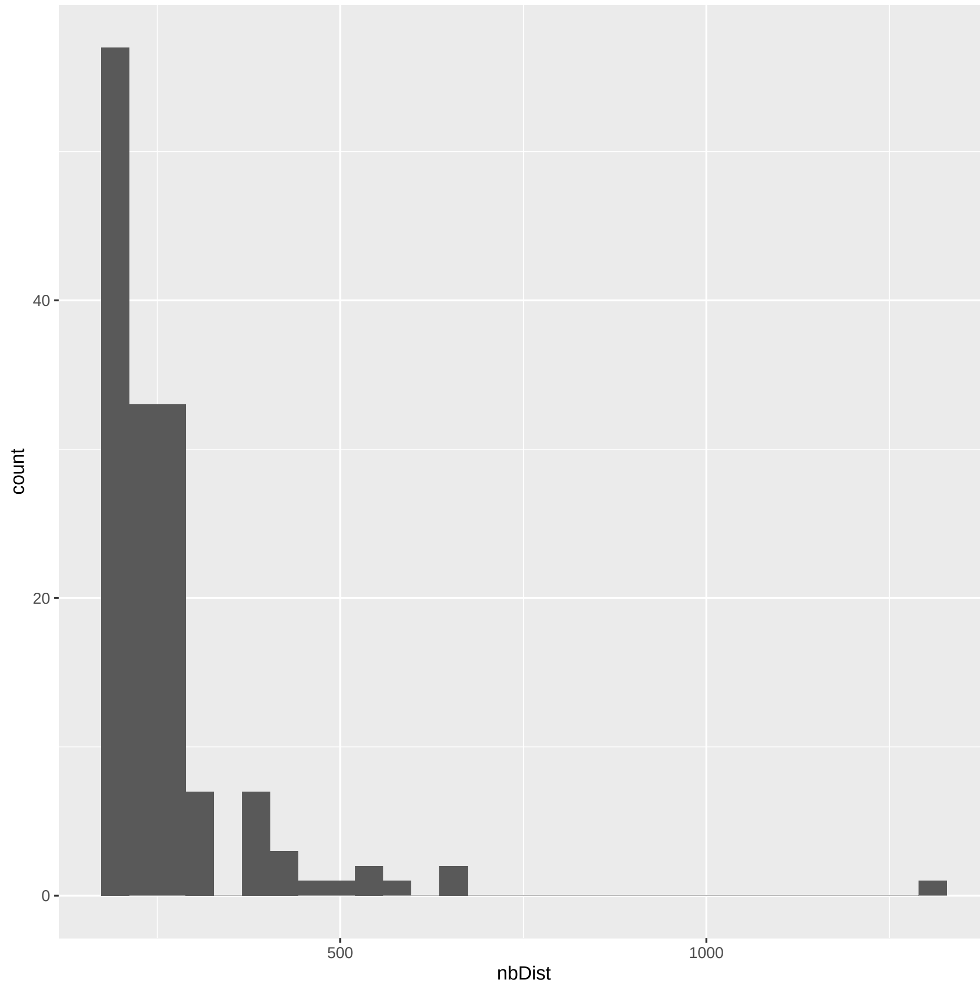
```
coords.hex <- st_coordinates(st_centroid(cellqt_NE_GL))
nc_sp <- as(cellqt_NE_GL, 'Spatial')
cellqt_NE_GL.Knbdist <- nbdists(neighbors, st_coordinates(st_centroid(cellqt_NE_GL)), longlat = NULL)

nbdists <- unlist(cellqt_NE_GL.Knbdist) %>% as.tibble()

nbdists %>% filter(nbdists < 125) %>% count()
```

```
# A tibble: 1 x 1
      n
  <int>
1     0
```

```
colnames(nbdists) <- "nbDist"

nbdists %>%
  ggplot(aes(nbDist)) +
  geom_histogram()

nbdists %>%
  filter(nbDist < 2000) %>%
  ggplot(aes(nbDist)) +
  geom_histogram()
```

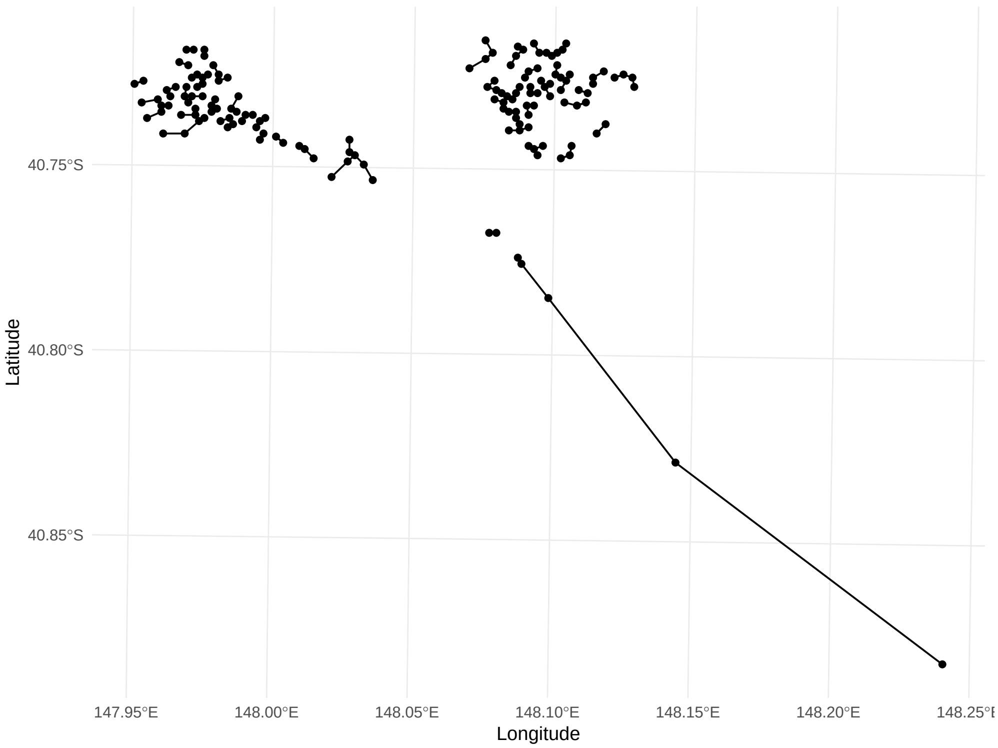


count
100
50
0
0
2500
5000
7500
10000
nbDist



Plot of nnb connections

```
neighbors_sf <- as(nb2lines(neighbors, coords = st_coordinates(st_centroid(cellqt_NE_GL))), 'sf' )
neighbors_sf <- st_set_crs(neighbors_sf, st_crs(cellqt_NE_GL))

ggplot(cellqt_NE_GL) +
  geom_sf(fill = 'salmon', color = 'white') +
  geom_sf(data = neighbors_sf) +
  geom_sf(data = st_centroid(cellqt_NE_GL)) +
  theme_minimal() +
  ylab("Latitude") +
  xlab("Longitude")
```



Export to geopackage

It is much easier to explore the data in QGIS. We export the selected cells as a hex layer and a point layer (centroid of the hex).

```
## Export cells to gpkg ####
pointqt <- st_centroid(cellqt_NE_GL)
outname.point <- sprintf("C:/Users/%s/University of Tasmania/IMAS-DiveFisheries - Assessments - Documen
st_write(pointqt, dsn = outname.point, layer = "pointqt", driver = "GPKG", append = F)
```

```
Deleting layer `pointqt' using driver `GPKG'
Writing layer `pointqt' to data source
  `C:/Users/jaimem/University of Tasmania/IMAS-DiveFisheries - Assessments - Documents/Assessments/GIS/
Writing 150 features with 46 fields and geometry type Point.
```

```
# Convert geometry to latitude and longitude to create dataframe
pointqt_df <- pointqt %>%
        st_transform(crs = st_crs(4326)) %>%
  sfheaders::sf_to_df(fill = T) %>%
  select(c(zone, blockno, subblockno, cell.ntile, oid, x, y)) %>%
  dplyr::rename('latitude' = y,
                'longitude' = x) %>%
```

```
 arrange((oid)) %>%
 mutate(site_order = row_number(),
        site = paste('GL', 2023, blockno, site_order, sep = '-'))

pointqt_sf  <-  pointqt_df %>%
  st_as_sf(coords = c('longitude', 'latitude'), crs = st_crs(4326))

# Export Excel file
write.xlsx(pointqt_df, sprintf("C:/Users/%s/University of Tasmania/IMAS-DiveFisheries - Assessments - Do
           sheetName = "Sheet1",
           col.names = TRUE, row.names = TRUE, append = FALSE)

# Export Excel file ready for GPX external file creation for plotter
# GDAL package for creating GPX files no longer supported in R
pointqt_gpx <- pointqt_df %>%
   select(site, longitude, latitude) %>%
   rename('waypointid' = site)

write.xlsx(pointqt_gpx, sprintf("C:/Users/%s/University of Tasmania/IMAS-DiveFisheries - Assessments - D
           sheetName = "Sheet1",
           col.names = TRUE, row.names = TRUE, append = FALSE)
```