Data Filtering Process, Condensed

Anna Steel

August 12, 2016

Filtering of VEMCO post-processed VPS positions

This code runs from the datafiles provided by VEMCO, and builds final datasets for analysis. It also periodcally outputs descriptive values (e.g.: N fish) to track how filtering alters the dataset.

The descriptive metrics produced include number of positions, number of fish, and positions per fish

The filtering includes the following stages, with more details in the scripts noted. * filters by HPE: see 'Exploration_DataFiltPrimary' for details * filters by speed: see 'Exploration_DataFiltSec1Speed' for details * filters for likely predators: see 'Exploration_DataFiltSec2Preds' for details * splits tracks with large gaps (for subsequent smoothing/rediscretization): see 'Exploration_TrackGapBias' for details

Remember: times are reported in UTC from vemco

Read in Data & Clean for proper dates and TagIDs

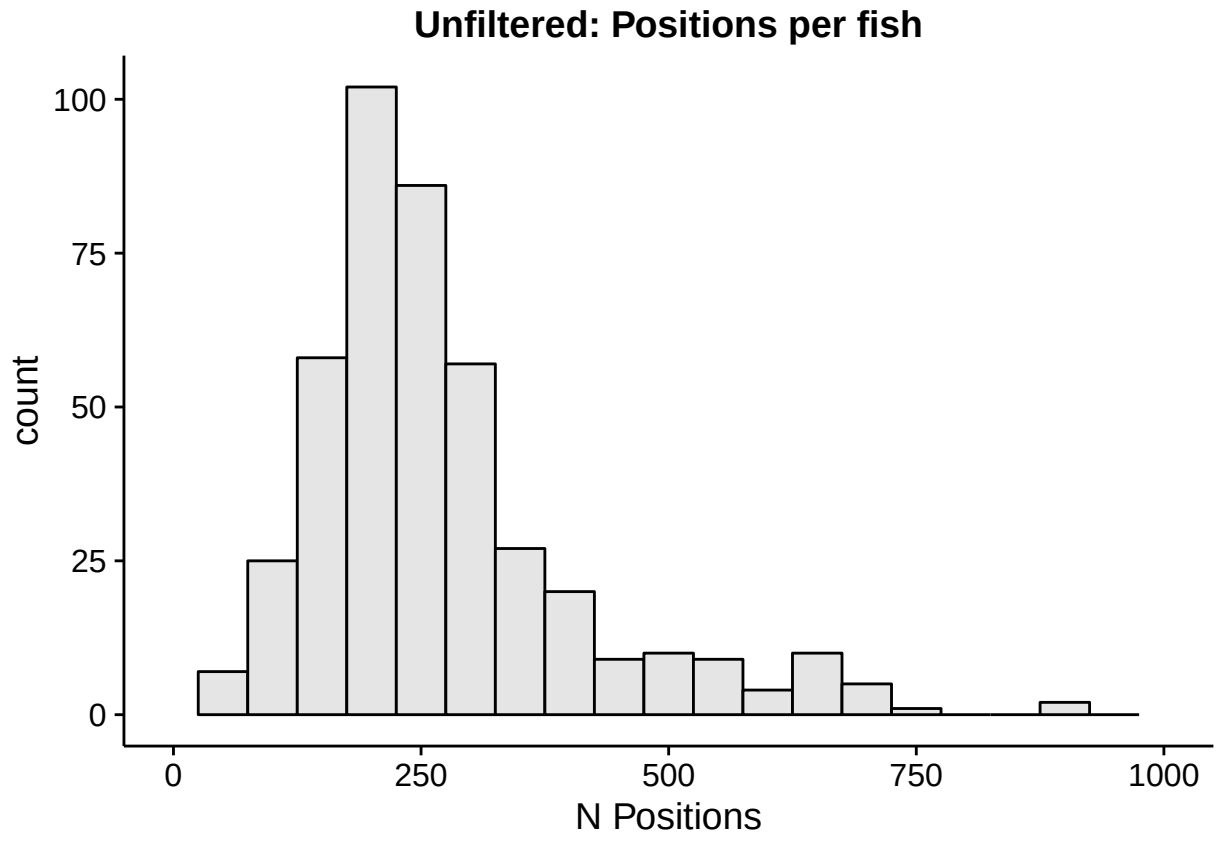
- Open script from Fremont16.Rproj in GitHub to ensure directories are correct
- Filter out tags in the 65xxx series
- Filter any tags detected outside of period of complete array (none this year)
- Add UTM coordinates
- Tabulate total fish, total positions, and total positions per fish

```
load("Maestros/alldf.RData")
options("digits.secs"=6)
alldf$Time <- as.POSIXct(as.character(alldf$Time), format="%Y-%m-%d %H:%M:%OS", tz = "GMT")

# remove fish tags in the 65xxx series (5 tags)
alldf <- alldf[alldf$Id<65000,]  # matches VEMCO

# incomplete array
alldfg <- alldf[alldf$Time > as.POSIXct("2016-02-109 14:00:00", tz="GMT"),]
 print(paste0(nrow(alldf) - nrow(alldfg)," positions removed due to incomplete VPS array"))
```

```
## [1] "0 positions removed due to incomplete VPS array"
```

```
# convert the Lat Long into UTMs using 'sp'
 alldf.sp <- SpatialPointsDataFrame(coords = alldf[,c("Longitude","Latitude")],
             data = alldf,
             proj4string=CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"))
       # confirmed string with VEMCO; the XY coords in azimuthal equal area
 options(digits=10)
 alldf.utm <- spTransform(alldf.sp,
                          CRS("+proj=utm +zone=10 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0
 alldf.utm@data[,c("east","north")] <- alldf.utm@coords

 alldf.utm = alldf.utm@data
```

```
## [1] "Prior to filtering, 122971 total positions in dataset"

## [1] "Prior to filtering, 433 individual fish positioned"

## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
## Warning: Removed 1 rows containing non-finite values (stat_bin).

## pdf
##   2

## [1] "Summary of N positions per fish, after reducing to applicable data"

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   37.0000  185.0000  238.0000  283.9977  317.0000 5577.0000
```

Primary HPE filter: <0.5 HPEs

```
## [1] "83658 positions"

## [1] "   68.03% of fish tag positions"

## [1] "433 individual fish"
```

```
## [1] "   100% of individual fish retained "
```

```
## [1] "Summary of N positions per fish, after filtering at HPE<0.5"
```

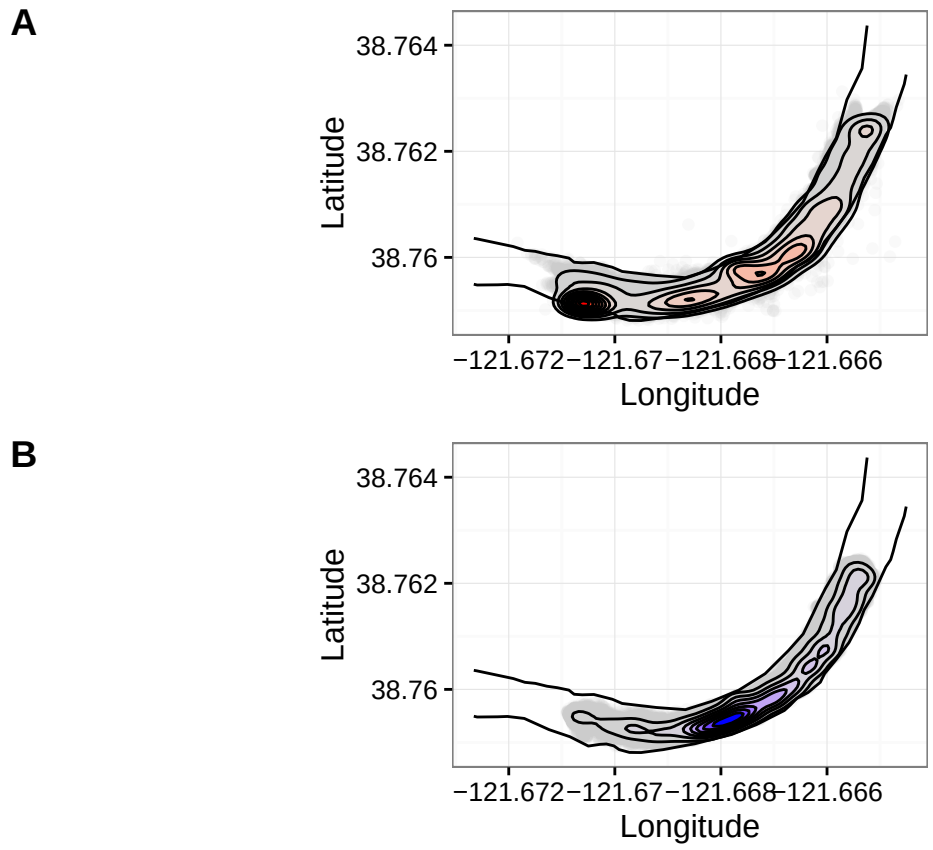
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  18.0000 124.0000 165.0000 193.2055 227.0000 642.0000
```

create plots of spatial errors, original, rejected, and retained

```
## OGR data source with driver: ESRI Shapefile
## Source: "GIS/2004_channel", layer: "2004_channel_freTightclipWGS84"
## with 2 features
## It has 1 fields
```

```
## Warning: Removed 13 rows containing non-finite values (stat_density2d).
```

```
## Warning: Removed 13 rows containing missing values (geom_point).
```




```
## Warning: Removed 13 rows containing non-finite values (stat_density2d).
```

```
## Warning: Removed 13 rows containing missing values (geom_point).
```

```
## pdf
##   2
```

```
save(reddf, file="Maestros/AllFishPrimaryFilt.RData")
```

End of Primary Filtering Process

Beginning of Secondary Filtering Process

Excessive Speeds Filtering

- Use primary filtered dataset created above
- Use adehabitatLT to calculate distance and speed between consecutive positions
- Idenitfy consecutive positions resulting in excessive speeds (top 1%, 7.7mps) — *consider other justfication for 'excessive' speed threshold?*
- Position only considered 'bad' if both the step to and the step from the position have excessive speeds.

```
# identify 'bad' positions (99%ile of step-speeds is ~7.7mps
red2$prevspd = lag(red2$dist)/lag(red2$dt)

red2$badpos <- 0
 red2$badpos[red2$spd_mps>7.7 & red2$prevspd>7.7] <- 1

# filter out bad positions
red3 = red2[red2$badpos==0,]

# recalculate speed and distance
red3.ltraj = as.ltraj(xy=red3[,c("east","north")], date=red3$date,
                      id=red3$Id, infolocs = red3[,c("Id","Hpes","east","north")])

red4 = ld(red3.ltraj)
red4$spd_mps = red4$dist / red4$dt

save(red4, file="Maestros/AllFish_FiltSec1Speed.RData")
```

```
## [1] "83609 positions"
```

```
## [1] "   67.99% of fish tag positions"
```

```
## [1] "433 individual fish"
```

```
## [1] "   100% of individual fish retained "
```

```
## [1] "Summary of N positions per fish, after filtering excessive speeds"
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  18.0000 124.0000 165.0000 193.0924 227.0000 642.0000
```

Remove fish which departed from and returned again to array

- Remove select bursts for fish returning to array after some time absent (here use 6 hrs)
- Identified fish track segments to remove manually in external code

```
dt_threshold = 6*3600  # 6 hours

dtcut = function(dt) { return (dt > dt_threshold) }

red4.ltraj = dl(red4)
red5.ltraj <- cutltraj(red4.ltraj, "dtcut(dt)", nextr=TRUE)
```

```
## Warning in cutltraj(red4.ltraj, "dtcut(dt)", nextr = TRUE): At least 3 relocations are needed for a l
##  1 relocations have been deleted
```

```
red5 = ld(red5.ltraj)

red5 = red5[!(red5$burst %in% c(36472.2, 36472.3, 36472.4, 36612.2, 36612.3)),]

save(red5, file="Maestros/AllFish_FiltSec2Pred.RData")
```

```
## [1] "83511 positions"

## [1] "   67.91% of fish tag positions"

## [1] "433 individual fish"

## [1] "   100% of individual fish retained "

## [1] "Summary of N positions per fish, after filtering return trips to array"

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  18.0000 124.0000 165.0000 192.8661 227.0000 642.0000
```

Remove fish demonstrating suspicious holding behavior

- See "Exploration_DataFilteringPreds" for more on these three tracks. **Should revisit this - these might be milling smolts, but might also be a predator**

```
red6 = red5[!(red5$Id %in% c(36379,36483,36675)),]
save(red6, file="Maestros/AllFish_FiltSec3Hold.RData")
```

```
## [1] "82061 positions"

## [1] "   66.73% of fish tag positions"

## [1] "430 individual fish"

## [1] "   99.3% of individual fish retained "

## [1] "Summary of N positions per fish, after filtering fish with suspicious holding behaviors"

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  18.0000 123.2500 165.0000 190.8395 226.0000 578.0000
```

Cut tracks into sub-bursts when gaps are > a selected threshold to avoid interpolating over long distances

These tracks will not be used for spatial aggregation, only for speed and turning angle analyses.

- For now I've used 50m gaps until I make time to evaluate this more deeply.
- This results in a few sub-bursts that retain <4 positions; these are automatically dropped by adehabitatLT (a total of 166 positions)
- Reference back to final few code chunks in "Exploration_TrackGapBias" for more details on how to select this threshold.

```
dist_threshold = 50
distcut = function(dist) { return (dist > dist_threshold) }

red6.ltraj = as.ltraj(xy=red6[,c("east","north")], date = red6$date, id = red6$Id, infolocs=red6[,c("Hp
red7.ltraj <- cutltraj(red6.ltraj, "distcut(dist)", nextr=TRUE)

red7 = ld(red7.ltraj)
  red7$spd_mps = red7$dist / red7$dt

 # N fish & total N positions
   print(paste0(nrow(red7)," positions"))
```

```
## [1] "81895 positions"
```

```
    print(paste0("   ",round(nrow(red7)/nrow(alldfg)*100,2), "% of fish tag positions"))
```

```
## [1] "   66.6% of fish tag positions"
```

```
   print(paste0(length(unique(red7$Id))," individual fish"))
```

```
## [1] "0 individual fish"
```

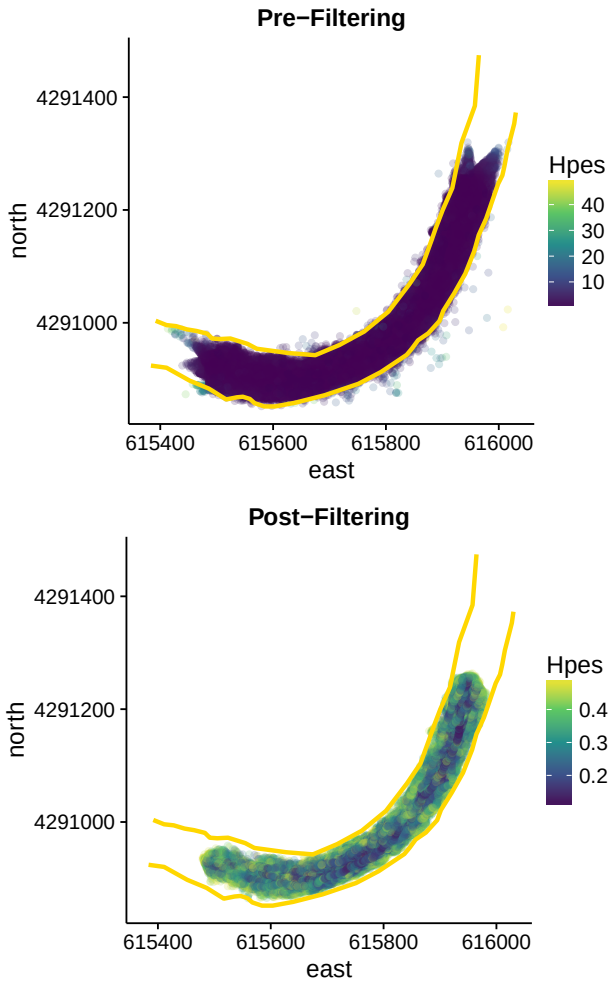
```
    print(paste0("   ",round((length(unique(red7$id))/length(unique(alldf$Id)))*100,1),"% of individual
```

```
## [1] "   99.3% of individual fish retained "
```

```
save(red7, file="Maestros/AllFish_FiltSec4Bursts.RData")
```

Plot of pre- & post-filtering positions

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:/Users/Anna/Documents/GitHub/Fremont16/GIS/2004_channel", layer: "2004_channel_freTightcl
## with 2 features
## It has 1 fields
```




```
## pdf
##   2
```

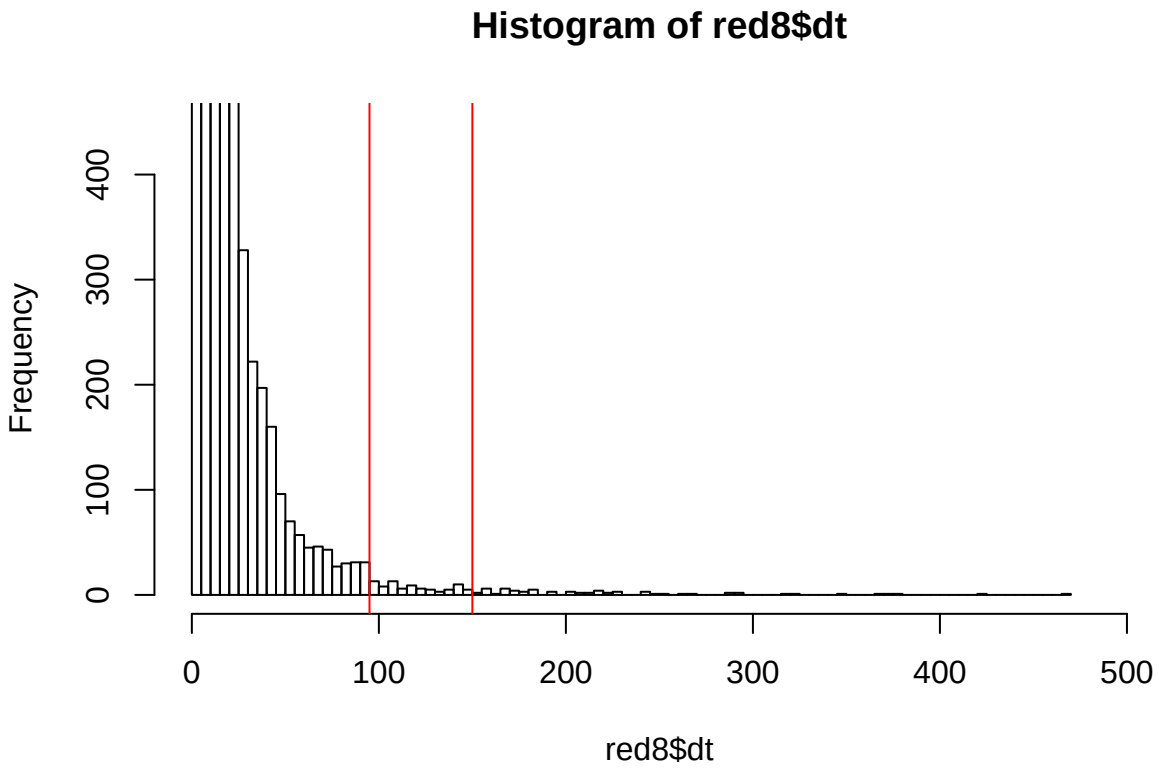
Remove tracks with >150 m gaps to avoid interpolating over long distances

These tracks will ONLY be used for spatial aggregation purposes (temp gaps redisc)

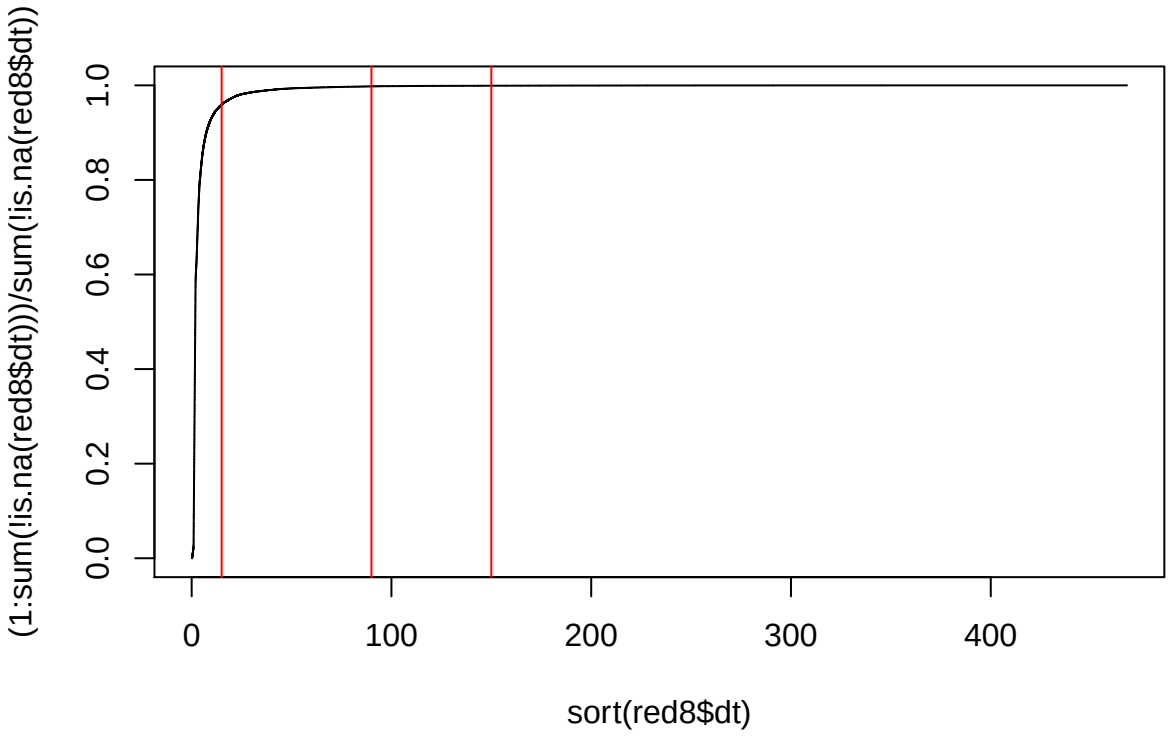
- For now I've used 50m gaps until I make time to evaluate this more deeply.
- This results in a few sub-bursts that retain <4 positions; these are automatically dropped by adehabitatLT (a total of 166 positions)
- Reference back to final few code chunks in "Exploration_TrackGapBias" for more details on how to select this threshold.

```
gap_threshold = 150
#gapcut = function(gap) { return (gap > gap_threshold) }

red6.ltraj = as.ltraj(xy=red6[,c("east","north")], date = red6$date, id = red6$Id, infolocs=red6[,c("Hp

 red8 = ld(red6.ltraj)
```

```r
# look at where 150 is in the histogram and ecdf, and consider other thresholds
hist(red8$dt, xlim=c(0,500), ylim=c(0,450), breaks=100)
 abline(v=95, col="red")
 abline(v=150, col="red")
```




```r
plot(sort(red8$dt), (1:sum(!is.na(red8$dt)))/sum(!is.na(red8$dt)), type="s")
 abline(v=15, col="red")
 abline(v=150, col="red")
 abline(v=90, col="red")
```




```
    # other thresholds might be better, but because we used 150 in 2015 we'll stick with that to make

  longgaps = red8[red8$dt>=gap_threshold,]; longgaps = longgaps[!is.na(longgaps$id),]
   gapids = unique(longgaps$id) #56 tags (13% of 430 tags detected in rel 1 & 2

  red9 = red8[!(red8$id %in% gapids),]

save(red9, file="Maestros/AllFish_FiltSec5Gaps.RData")
```