

# 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 periodically 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=0,0,0"))
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"

## [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: <1.0 HPEs

```

## [1] "106529 positions"

## [1] " 86.63% 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<1"

##      Min.    1st Qu.    Median    Mean    3rd Qu.    Max.
## 28.0000 165.0000 212.0000 246.0254 283.0000 2015.0000

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
- Identify consecutive positions resulting in excessive speeds (top 1%, 7.7mps) — *consider other justification 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] "106339 positions"

## [1] " 86.47% 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.
## 28.0000 165.0000 212.0000 245.5866 283.0000 2013.0000

```

## 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)

red5 = ld(red5.ltraj)

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

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

## [1] "106125 positions"

```

```

## [1] " 86.3% 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.
## 28.0000 165.0000 212.0000 245.0924 283.0000 2013.0000

```

## 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] "102736 positions"

## [1] " 83.54% 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.
## 28.0000 164.2500 210.0000 238.9209 281.7500 660.0000

```

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

- 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 103 positions)
- Reference bac to final few code chunks in “Exploration\_TrackGapBias” for more details on how to select this threshold.

```

dist_threshold = 50
gapcut = function(dist) { return (dist > dist_threshold) }

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

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

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_frtightcl"
## with 2 features
## It has 1 fields
```

