# 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
- Tabulate same metrics for releases 1&2 and 3-5 seperately, b/c of differences in flow during those periods, and because they were processed seperately by VEMCO (even though I'll lump them here for filtering and analysis)

```
alldf <- readRDS("Maestros/AllPos1to5.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("2016-02-09 14:00:00", tz="GMT"),]
#   print(paste0(nrow(alldf) - nrow(alldfg)," 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, 155741 total positions in dataset"
```

```
## [1] "Prior to filtering, 644 individual fish positioned"
```

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

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   19.0000  150.0000  204.0000  241.8339  276.2500 5577.0000
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:adehabitatLT':
##
##     id
```

```
## The following object is masked from 'package:adehabitatMA':
##
##     join
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## [1] "Prior to filtering, 123275 total positions in Releases 1 & 2"
```

```
## [1] "Prior to filtering, 434 individual fish positioned from Releases 1 & 2"
```

```
## [1] "Summary of N positions per fish from Releases 1 & 2, after reducing to applicable data"
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   19.0000  184.2500  238.5000  284.0438  318.5000 5577.0000
```

```
## [1] "Prior to filtering, 32466 total positions in Releases 3, 4, & 5"
```

```
## [1] "Prior to filtering, 210 individual fish positioned from Releases 3, 4, & 5"
```

```
## [1] "Summary of N positions per fish from Releases 3, 4, & 5, after reducing to applicable data"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.00  119.25  156.00  154.60  188.00  312.00
```

Note: these values appear to differ from those presented in the VEMCO report. These values are based on release dates, and the vemco reports are based on detection dates. There were two fish from release 2 which were detected after 2016-03-09 00:00:00 GMT, for a total of 304 positions. One was only detected after that date, and the other was detected both before and after.

### Primary HPE filter: <0.5 HPEs

**To see an assessment of error of sync tags, refer to full script "Exploration_DataFiltPrimary.R"**

```
## [1] "104712 positions"
```

```
## [1] "   67.23% of fish tag positions"
```

```
## [1] "644 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.
##   3.0000  97.0000 139.0000 162.5963 195.0000 642.0000
```

```
saveRDS(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

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

```
## [1] "104501 positions"
```

```
## [1] "   67.1% of fish tag positions"
```

```
## [1] "644 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.
##   3.0000  96.7500 139.0000 162.2686 195.0000 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, 36612.2, 38902.2)),]

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

```
## [1] "104349 positions"
```

```
## [1] "   67% of fish tag positions"
```

```
## [1] "644 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.
##   3.0000  96.0000 139.0000 162.0326 194.2500 642.0000
```

### Remove fish demonstrating suspicious holding behavior

- See "Exploration_DataFilteringPreds" for more on these three tracks.
- Spoke with Paul Stumpner (USGS) and he confirmed that the velocities in this area are too high for small smolts to hold and mill like this; indicative of a predator.

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

```
## [1] "102899 positions"
```

```
## [1] "   66.07% of fish tag positions"
```

```
## [1] "641 individual fish"
```

```
## [1] "   99.5% 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.
##   3.0000  96.0000 139.0000 160.5289 194.0000 578.0000
```

### 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("Hp
red7.ltraj <- cutltraj(red6.ltraj, "gapcut(dist)", nextr=TRUE)

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

saveRDS(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
```