Rediscretization_Temporal

Anna Steel

September 6, 2016

Rediscretization of Tracks - 20 seconds between positions

- Using primary and secondary filtered data to rediscretize tracks for further analysis
- Tracks with gaps > 150 m have been removed entirely from the analysis. =/
- this threshold can be altered in "Final_Filtering.Rmd" if desired
- Before redistretizing, remove bursts with < 10 positions (too few to rediscretize in adehabitatLT)
- also note that the interval of 20 seconds was selected to be consistent with the 2015 analysis; another script will discretize at 2 seconds to be more consitent with ELAM outputs and USGS analysis

```
## [1] 76269    15
```

```
## [1] 374
```

```
## [1] 374
```

```
## [1] 203.92780748663102
```

```
## [1]  22 578
```

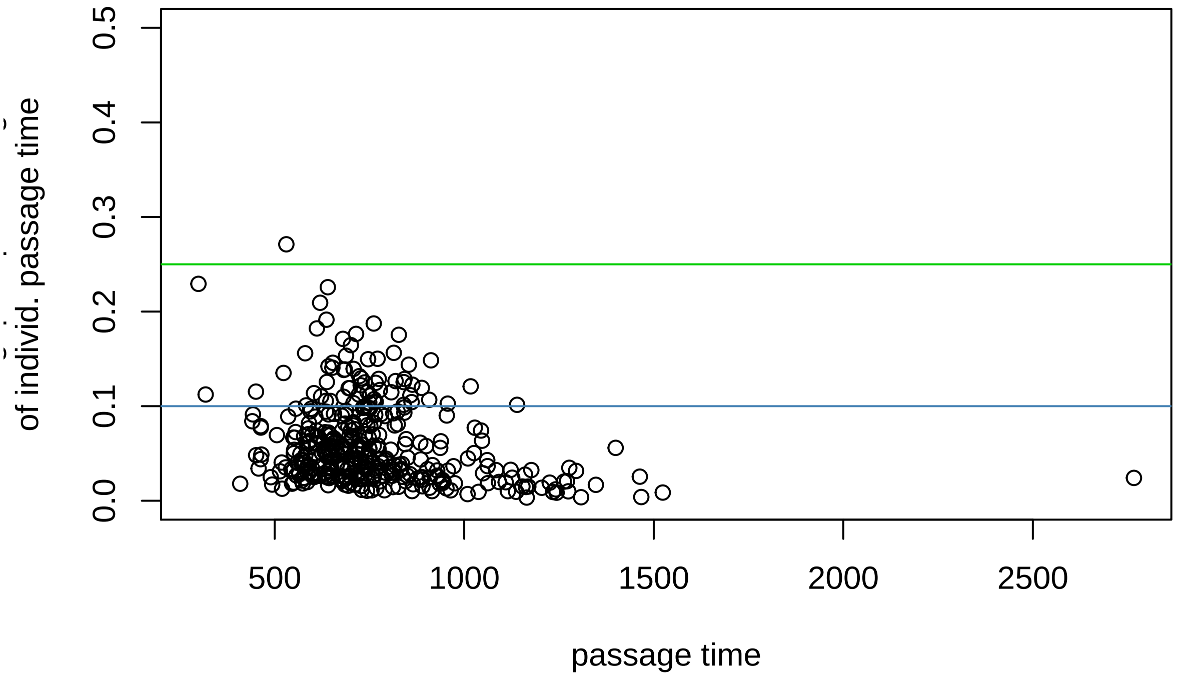
Stagger first detection to a random location within the first 100m of start of the array

```
   # stagger first detection across all fish
ndetects = summarize(group_by(red9, id), ndet = n())
quantile(ndetects$ndet, .05)  # fewest 5% of fish have >86 points before discretization
```

```
##                 5%
## 86.650000000000006
```

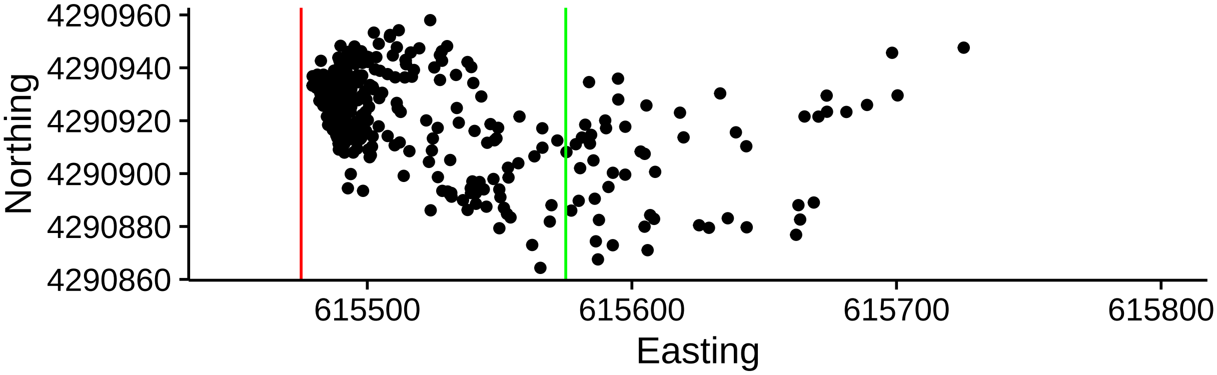
```
  # look at gaps in detection times across the first 25 detections for each fish
detect.gaps = red9 %>%
   group_by(id) %>%
   mutate(passage.time = sum(dt, na.rm=T)) %>%
   slice(1:25) %>%
   ungroup() %>%
   group_by(id) %>%
   summarize(mean.gap = mean(dt, na.rm=T), median.gap= median(dt, na.rm=T), max.gap=max(dt, na.rm=
detect.gaps = as.data.frame(detect.gaps)

  # look at how big the gaps are in relation to total passage time - are the fish half way through
detect.gaps$percmaxgap = detect.gaps$max.gap / detect.gaps$passage.time
plot(detect.gaps$percmaxgap ~ detect.gaps$passage.time, ylim=c(0,.5), xlab="passage time", ylab="ma
 abline(h=.1, col="steelblue")  # above line, max gap is > 1/10 of total passage time
 abline(h=.25, col="green3")  # above line, max gap is > 1/4 of total passage time
```



```
 # plot first detection of fish
firsts = as.data.frame(red9 %>% group_by(id) %>% slice(1))

ggplot(data=firsts, aes(y=y, x=x)) + geom_point() +
  coord_fixed() +
  xlim(c(615450, 615800)) +
  ylab("Northing") + xlab("Easting") +
  geom_vline(xintercept = 615475, col="red") +
  geom_vline(xintercept = 615575, col="green")
```



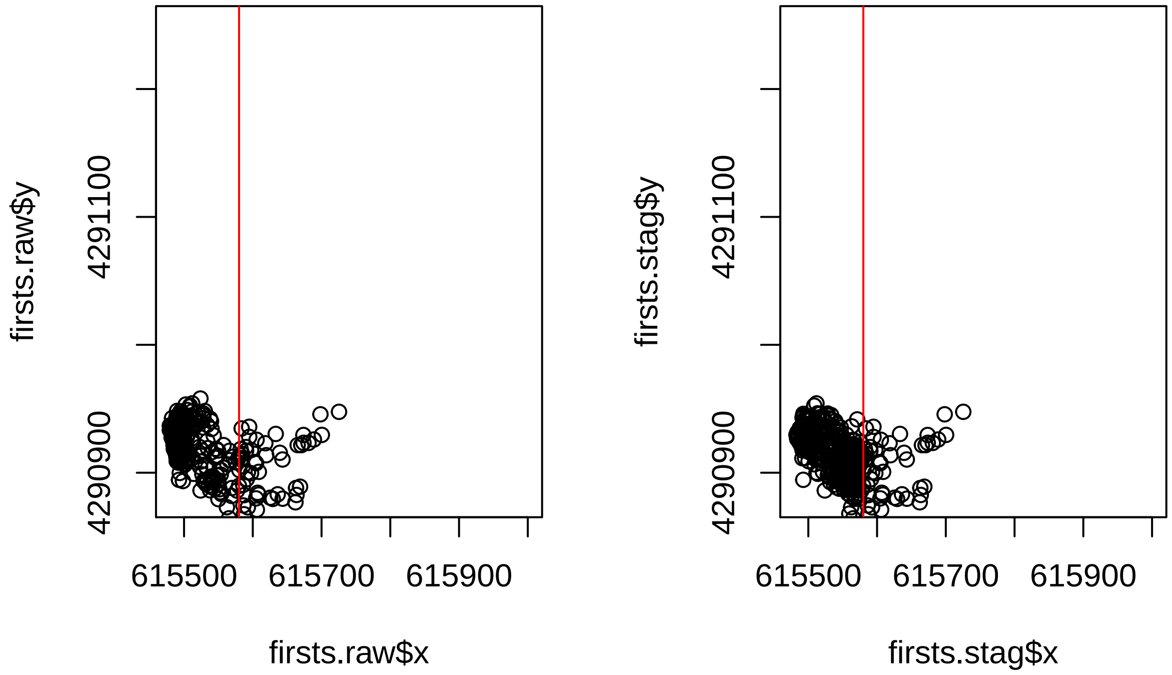
```
 # if first detection has an x value greater than 615575 then keep it (>100m from start)
 #    - there are 64 fish for which this applies
 # otherwise, pick randomly from the points before that location

 # pull number of points for each id that are before the threshold
n.prethres = red9 %>%
   filter(x<615575) %>%
   group_by(id) %>%
   summarize(n.pre = n())
n.prethres = as.data.frame(n.prethres)

  # add column with the first detection selected for use for that id
  samplefunc = function(x) sample(1:x,1)
  set.seed(23)
  n.prethres$first.det = unlist(lapply(n.prethres$n.pre, samplefunc))

 # subset the original data frame using the randomly sampled first detection
stag.prep = red9 %>%
   group_by(id) %>%
   mutate(detect.num = 1:n())
stag.prep = as.data.frame(stag.prep)
reddf.stag = merge(stag.prep, n.prethres, all.x=T)
 reddf.stag[is.na(reddf.stag$first.det),"first.det"] <- 1  # if the first detection was past thres
 reddf.stag = reddf.stag[order(reddf.stag$id, reddf.stag$detect.num),]
 reddf.stag = reddf.stag[reddf.stag$detect.num >= reddf.stag$first.det,]
```

```
    # compare first detection location before and after staggering
   windows(); par(mfrow=c(1,2))
   firsts.raw = as.data.frame(red9 %>% group_by(id) %>% slice(1))
   plot(firsts.raw$y ~ firsts.raw$x, xlim=c(615480,616000), ylim=c(4290880, 4291250))
    abline(v=615580, col="red")
   firsts.stag = as.data.frame(reddf.stag %>% group_by(id) %>% slice(1))
   plot(firsts.stag$y ~ firsts.stag$x, xlim=c(615480,616000), ylim=c(4290880, 4291250))
    abline(v=615580, col="red")
```



```
    # looks great! The blob of detections right at the beginning is now spread out across the first 1
    reddf.stag$n.pre <- NULL
    reddf.stag$detect.num <- NULL
```

Make an ltraj object

```
red9.ltraj = as.ltraj(xy=reddf.stag[,c("east","north")], date=reddf.stag$date,
                      id=factor(reddf.stag$id), burst = factor(reddf.stag$burst),
                      infolocs=reddf.stag[,c("Hpes","east","north")])
```

Discretize in Time

```
 # discretize in time
 red9.trdz = ld(redisltraj(red9.ltraj, u=20, type="time", nnew=50))
  red9.trdz$run = "LFC" # creates a common grouping variable to make UD with all points
  red9.trdz=red9.trdz[order(red9.trdz$id,red9.trdz$date),]

# recalculate migration speed
  red9.trdz$spd_mps = red9.trdz$dist / red9.trdz$dt
```

And finally, output the general metrics about the remaining dataset

```
   dim(red9.trdz) # 14466 detections after discretization
```

```
## [1] 13815    31
```

```
   length(unique(red9.trdz$id))  # 374
```

```
## [1] 374
```

```
   ndetects.discr = summarize(group_by(red9.trdz, id), ndet = n())
     mean(ndetects.discr$ndet) # 38.68 per fish
```

```
## [1] 36.938502673796791
```

```
     range(ndetects.discr$ndet) # ranges from 15 - 139
```

```
## [1]  15 136
```

```
   max(red9.trdz$spd_mps, na.rm=T)  # 4.81 mps
```

```
## [1] 3.5416678741730152
```