Getting started with summarizing Motus .Rds detection files

Zoe Crysler and Tara Crewe

September 1, 2016

This document provides some R code and associated output for Motus users to get started with their Motus detections data.

Load R Packages

First, we load require R packages. If you do not have these packages installed, you will need to first install them using, e.g., > install.packages("plyr").

```
require(plyr)
require(ggplot2)
require(ggmap)
require(RgoogleMaps)
require(lubridate)
```

Import and manage RDS tag detection file

When the data are imported, the tag "id" is an integer, so we change the format to a factor. For ease of plotting later, we also order site factor levels based on latitude, then sort the full dataframe by time stamp so that movement tracks are connected in chronological order. Here we import and summarize a subset of data on shorebirds tagged in James Bay, Ontario, Canada, and is provided for this purpose courtesy of the James Bay Shorebird Monitoring Project - Environment and Climate Change Canada. Permission to use the data for any other purpose should be requested from the data owners.

```
tags <- readRDS("F:/RScripts/Data/2015_JamesBay_globaltags_small.rds")

tags$id <- as.factor(tags$id)
tags <- within(tags, site <- reorder(site, (lat)))
tags <- tags[order(tags$ts),]
```

Playing with dates

Dates can be confusing to work with, and there are a variety of ways to deal with them. Here we use strptime() to create a date variable by removing the time (hour, minutes, seconds) from the time stamp. The plyr() package can be used to round a time stamp to the nearest seconds if, for example, you want to group detections into time bins. Note also that

RDS files generally come in GMT, and can be summarized using this time zone, or adjusted to your current time zone (we leave it as is here).

```
tags$date <- strptime(tags$ts, "%Y-%m-%d")
tags$date <- as.POSIXct(tags$date) ## removes time from ts

tags$ts_10 <- plyr::round_any(tags$ts, 10) ## rounds time stamp to nearest 10
second

difftime(max(tags$ts), min(tags$ts), units = "days") # number of days between
first and last detection

## Time difference of 75.43757 days
```

Now, a few examples of extracting information on dates using lubridate(), and checking to see if the times that two individuals spent at the Longridge site overlapped.

```
tags$year <- year(tags$ts)
tags$month <- month(tags$ts)
tags$mday <- day(tags$ts) # day of the month
tags$wday <- wday(tags$ts) # weekday
tags$yday <- yday(tags$ts) # day of the year
tags$hour <- hour(tags$ts)
tags$minute <- minute(tags$ts)
tags$second <- second(tags$ts)

# get time intervals that two birds were at Longridge, and see if the time
intervals overlapped:

int.174 <- with(subset(tags, id == 171 & site == "Longridge"),
interval(min(ts), max(ts)))
int.376 <- with(subset(tags, id == 378 & site == "Longridge"),
interval(min(ts), max(ts)))
int_overlaps(int.174, int.376) # these birds did not overlap at Longridge

## [1] FALSE
```

Summary statistics

Tags detected at each site

Create a simple table to show which tags (columns) were detected at each site (rows).

```
with(tags, table(site, id))

##                 id
## site             171  173  174  176  177  178  179  180  181  182  185
##   FI               0    0    0    0    0    0    0    0   80    0    0
##   Bull             0    0    0    0    0    0    0    0   53    0    0
##   PrimeHook        0    0    0    0    0    0    0    0   68    0    0
##   BombayHook       0    0    0    0    0    0    0    0  117    0    0
##   HOLG             0    0    0    0    0    0    0    0   17    0    0
##   Rutgers          0    0    0    0    0    0    0    0   24    0    0
##   bise             0    0    0    0    0    0    0    0    0    0    0
##   trus             0    0    0    0    0    0    0    0    0    0    0
##   Sugarloaf        0    0    0    0    0    0    0    0    0    0    0
##   Jordan           0    0    0    0    0    0    0    0   30    0    0
##   PortMaitland     0    0    0    0    0    0    0    0   38    0    0
##   EagleHead        0    0    0    0    0    0   43    0    0    0    0
##   BrierIsland      0    0    0    0   32    0    0    0    0    0    0
##   SWHead           0    0    0    0   63    0    0    0    0    0    0
##   LOOKSPT          0    0    0    0    0    0    0    0  107    0    0
##   SwallowTail      0    0    0    0    3    0    0    0    0    0    0
##   PointLepreau     0    0    0    0    0    0    7    0    0    0    0
##   NewHarbour       0    0    0    0    0    0    0    0    0    0    0
##   MtThom           0    0    0    0    0    0    0    0    0    0    0
##   Estimauville     0    0    0    0   28    0    0    0   37    0    0
##   StDenisSurMer    0    0    0    0    0    0    0    0    0    0    0
##   Washkagou        0    0    0   50   24   41    0    0   18    0    0
##   Netitishi        0    0   91  293  229   59 1005    0  416    0    0
##   NP_SRX           0    0    3    0    3    3   13    8    4    4   11
##   NorthPoint     405   47    0   18    0    0   24   43    0    0    6
##   Piskwamish      36   68    0    0    0    0    0    0    0    0    0
##   Longridge      366    0    0    0    0    0    0    0    0    0    0
##   sach             0    0    0    0    0    0    0    0    0    0    0
##   BennettMeadow    0    0    0    0    0    0    0    0    0    0    0
##   FortRiver        0    0    0    0    0    0    0    0    0    0    0
##   Hogback          0    0    0    0    0    0    0    0    0    0    0
##   MountToby        0    0    0    0    0    0    0    0    0    0    0
##   Shelburne        0    0    0    0    0    0    0    0    0    0    0
##                 id
## site            186  376  378  379  395
##   FI               0    0    0   31    0
##   Bull             0    0    0    5    0
##   PrimeHook        0    0    0    0    0
##   BombayHook       0    0    0    0    0
##   HOLG             0    0    0    0    0
##   Rutgers          0    0    0    0    0
##   bise             0    0   45    0    0
##   trus             0    0   15    0    0
##   Sugarloaf        0    0    8    0    0
##   Jordan           0    0    0    0    0
##   PortMaitland     0    0    0    0    0
##   EagleHead        0    0    0    0    0
##   BrierIsland      0    0    0    0    0
##   SWHead           0    0    0    0    0
##   LOOKSPT          0    0    0    0    0
##   SwallowTail      0    0    0    0    0
##   PointLepreau     0    0    0    0    0
##   NewHarbour        0    0    0    0   21
##   MtThom            0    0    0    0   40
##   Estimauville      0    0    0    0    0
##   StDenisSurMer     0    0    0    0   12
##   Washkagou         0    0    0    0   22
##   Netitishi         7    0    0    0    0
##   NP_SRX           11    0    0    0    0
##   NorthPoint     245    0  201   46   19
##   Piskwamish        0   15  206    0    0
##   Longridge         0    0  131    0   14
##   sach              0    0    0    0    0
##   BennettMeadow     0    0   32    0    0
##   FortRiver         0    0   26    0    0
##   Hogback           0    0   96    0    0
##   MountToby         0    0   10    0    0
##   Shelburne         0    0    5    0    0
```

Number of tags and detections at each site

To summarize the number of tags and detections at each site, we use ddply() to create a new dataframe 'sum1'with two new columns: 'num.birds', which is the number of unique tag id's detected at each site; and 'num.det', which is the total number of detections for each site across all tags. You can then use view() to have the resulting data pop up in a new window in RStudio, or, as done here, simply type 'sum1' to print it to the R console.

```
sum1 <- ddply(tags, .(site),
               summarize,
               num.birds = length(unique(fullID)),
               num.det = length(ts))
#View(sum1)
sum1
```

```
##             site num.birds num.det
## 1             FI         2     111
## 2           Bull         2      58
## 3      PrimeHook         1      68
## 4     BombayHook         1     117
## 5           HOLG         1      17
## 6        Rutgers         1      24
## 7           bise         1      45
## 8           trus         1      15
## 9      Sugarloaf         1       8
## 10        Jordan         1      30
## 11  PortMaitland         1      38
## 12     EagleHead         1      43
## 13   BrierIsland         1      32
## 14        SWHead         1      63
## 15       LOOKSPT         1     107
## 16   SwallowTail         1       3
## 17  PointLepreau         1       7
## 18     NewHarbour         1      21
## 19         MtThom         1      40
## 20   Estimauville         2      65
## 21 StDenisSurMer         1      12
## 22      Washkagou         5     155
## 23      Netitishi         7    2100
## 24         NP_SRX         9      60
## 25     NorthPoint        10    1054
## 26     Piskwamish         4     325
## 27      Longridge         3     511
## 28  BennettMeadow         1      32
## 29      FortRiver         1      26
## 30        Hogback         1      96
## 31      MountToby         1      10
## 32      Shelburne         1       5
```

We then use ggplot() to visualize these summaries.

Figure 1. Bar plot showing the total number of detections at each site, across all tags.

```
p <- ggplot(sum1, aes(site, num.det))
p + geom_bar(stat = "identity") + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  ## make x-axis
labels on a 45Âº angle to read more easily
  labs(x= "Site", y = "Total detections") ## changes x- and y-axis label
```

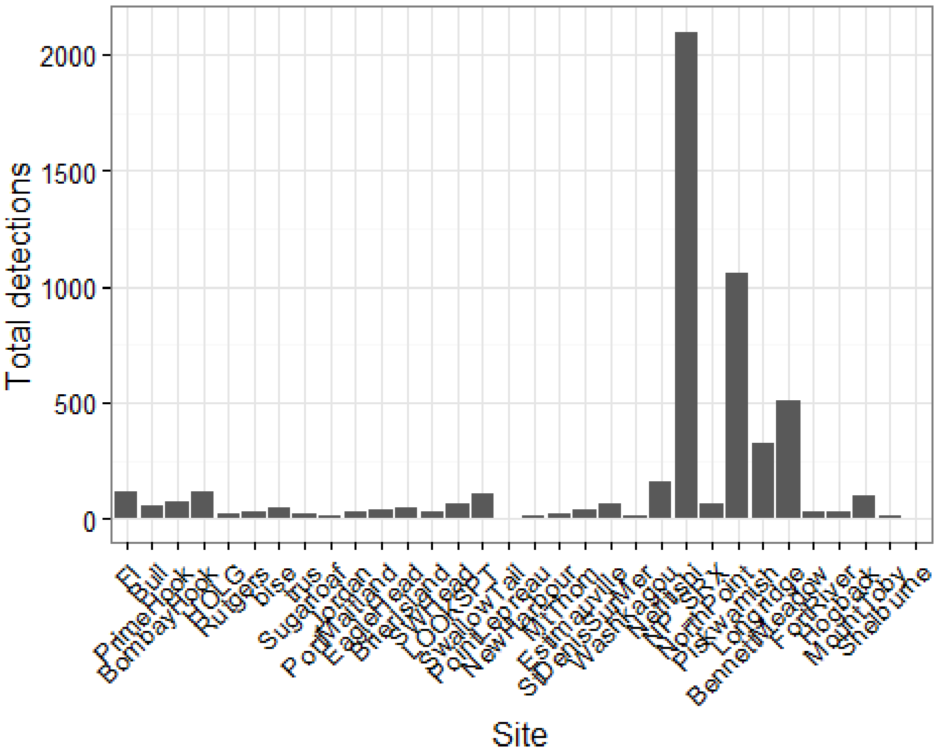
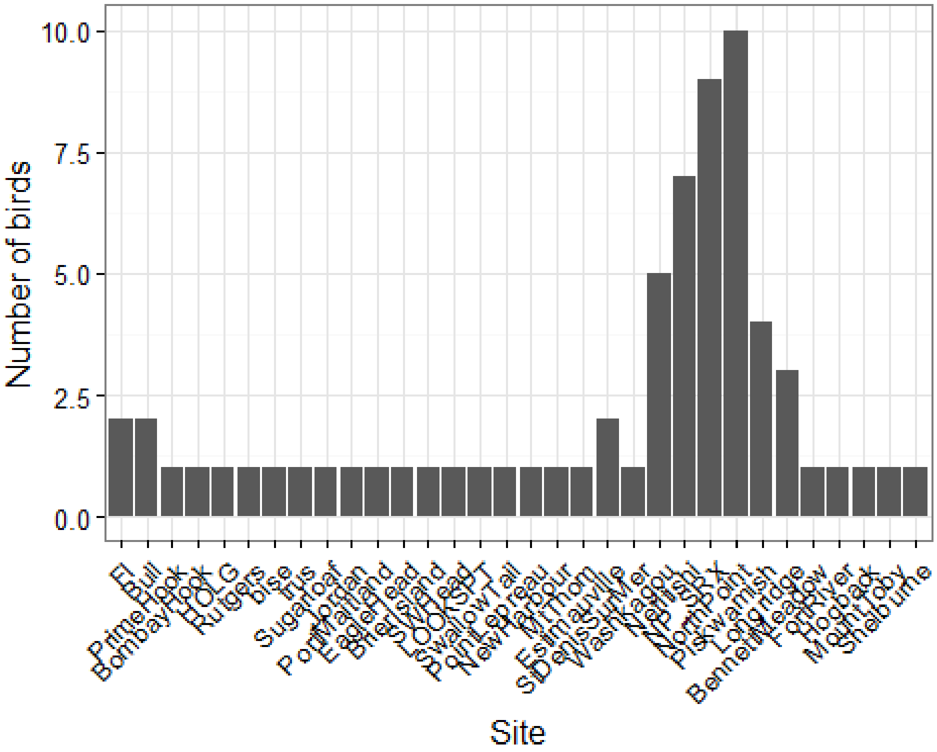


Figure 2. Bar plot showing the number of unique tags detected at each site.

```
p <- ggplot(sum1, aes(site, num.birds))
p + geom_bar(stat = "identity") + theme_bw() + ## creates bar plot by site
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + ## make x-axis
labels on a 45Âº angle to read more easily
  labs(x= "Site", y = "Number of birds") ## changes x- and y-axis label
```



Number of tags and detections each day at a specified site (Longridge)

To summarize the number of tags and detections at a specific site across days, we use the date variable (with hour and minute removed) to allow grouping by day. We then use ddply() to summarize a subset of the dataset 'tags' (site = "Longridge") and create a new dataframe 'sum2' which contains two new columns: 'num.birds', which gives the number of unique tag id's for each site and day; and 'num.det', which gives the total number of detections by site and day.

```
sum2 <- ddply(subset(tags, site == "Longridge"), .(site, date),
                summarize,
                num.birds = length(unique(fullID)),
                num.det = length(ts))

#View(sum2) ## view output in separate RStudio window
sum2

##          site       date num.birds num.det
## 1   Longridge 2015-08-19         1      19
## 2  Longridge 2015-08-20         1      26
## 3  Longridge 2015-08-21         1      35
## 4  Longridge 2015-08-22         1      21
## 5  Longridge 2015-08-23         1      20
## 6  Longridge 2015-08-24         1      19
## 7  Longridge 2015-08-25         1      49
## 8  Longridge 2015-08-27         1       5
## 9  Longridge 2015-08-28         1      94
## 10 Longridge 2015-08-29         1      62
## 11 Longridge 2015-08-30         1      15
## 12 Longridge 2015-08-31         1       1
## 13 Longridge 2015-09-12         1     125
## 14 Longridge 2015-09-13         2      12
## 15 Longridge 2015-09-14         1       8
```

Again, we use ggplot to visualize the summaries.

Figure 3. Bar plot of the total number of detections each day at Longridge, across all tags.

```
p <- ggplot(sum2, aes(date, num.det))
p + geom_bar(stat = "identity") + theme_bw() + ## creates bar plot by site
  labs(title = "Longridge", x= "Date", y = "Total detections") ## changes
title, x- and y-axis label
```

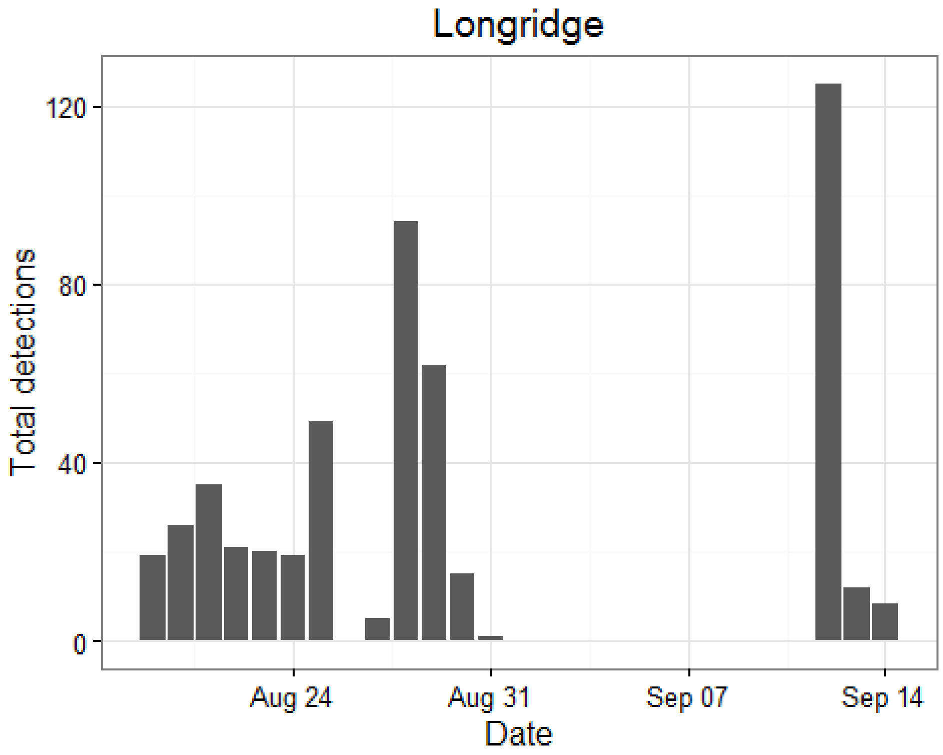
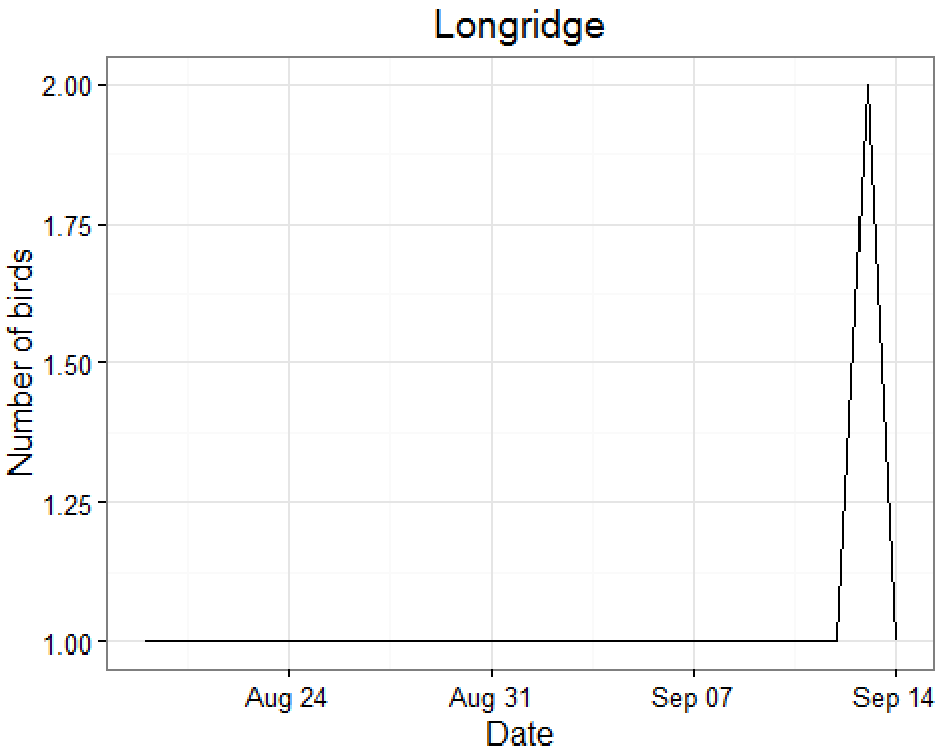


Figure 4. Line plot showing the total number of tags (birds) detected each day at the Longridge site.

```
p <- ggplot(sum2, aes(date, num.birds))
p + geom_line() + theme_bw() + ## creates line graph by site
  labs(title = "Longridge", x= "Date", y = "Number of birds") ## changes 
title, x- and y-axis label
```



Timing of detections at each site

We may be interested in the timing of detections at each site. Here we ddply to summarize min, max and the time between first and last detections at each site (across all tags).

```
sum3 <- ddply(tags, .(site), ## applies below function  to dataframe "tags" 
grouped by "site"
               summarize,
               first.ts = min(ts), ## creates new column "first.ts" that 
isolates first detection at each site
               last.ts = max(ts), ## creates new column "last.ts" that 
isolates last detection at each site
               range = paste(first.ts, last.ts, sep = " - "), ## creates new 
column of date range by pasting first and last detection timestamp
               tot.ts = difftime(last.ts, first.ts, units = "days")) ## 
determines length of time between first and last detections by number of days
#View(sum3)
sum3
```

```
##             site            first.ts             last.ts
## 1             FI 2015-09-19 01:50:22 2015-10-26 18:49:01
## 2           Bull 2015-09-19 01:59:52 2015-10-26 17:50:36
## 3      PrimeHook 2015-09-13 04:58:46 2015-09-13 05:01:49
## 4     BombayHook 2015-09-12 10:45:07 2015-09-12 10:52:20
## 5           HOLG 2015-09-12 04:40:34 2015-09-12 04:41:29
## 6        Rutgers 2015-09-11 13:28:47 2015-09-11 13:31:07
## 7           bise 2015-10-26 14:00:28 2015-10-26 15:06:15
## 8           trus 2015-10-26 14:13:14 2015-10-26 14:35:13
## 9      Sugarloaf 2015-10-26 12:29:59 2015-10-26 12:37:16
## 10        Jordan 2015-09-02 10:53:18 2015-09-02 10:57:21
## 11  PortMaitland 2015-09-02 09:50:24 2015-09-02 09:54:22
## 12     EagleHead 2015-09-08 15:58:13 2015-09-08 16:02:41
## 13   BrierIsland 2015-09-02 10:03:19 2015-09-02 10:06:09
## 14        SWHead 2015-09-02 09:06:13 2015-09-02 09:14:27
## 15       LOOKSPT 2015-09-02 07:33:09 2015-09-02 07:44:13
## 16   SwallowTail 2015-09-02 09:21:27 2015-09-02 09:21:39
## 17  PointLepreau 2015-09-08 13:45:14 2015-09-08 13:46:08
## 18    NewHarbour 2015-10-20 17:43:24 2015-10-20 17:46:58
## 19        MtThom 2015-10-20 16:35:38 2015-10-20 16:42:25
## 20  Estimauville 2015-09-02 02:44:50 2015-09-02 04:03:54
## 21 StDenisSurMer 2015-10-20 10:20:33 2015-10-20 10:22:20
## 22     Washkagou 2015-08-18 17:10:31 2015-10-20 02:24:52
## 23     Netitishi 2015-08-14 12:01:58 2015-09-07 21:09:06
## 24        NP_SRX 2015-08-12 08:18:55 2015-09-02 14:41:39
## 25    NorthPoint 2015-08-12 19:32:59 2015-10-25 18:17:05
## 26    Piskwamish 2015-08-19 08:45:38 2015-10-20 15:05:47
## 27     Longridge 2015-08-19 09:30:47 2015-09-14 17:47:31
## 28 BennettMeadow 2015-10-26 07:12:04 2015-10-26 07:30:49
## 29     FortRiver 2015-10-26 08:31:28 2015-10-26 08:41:58
## 30       Hogback 2015-10-26 06:35:53 2015-10-26 06:59:29
## 31     MountToby 2015-10-26 07:27:06 2015-10-26 07:29:31
## 32     Shelburne 2015-10-26 07:19:49 2015-10-26 07:22:44
##                                        range            tot.ts
## 1  2015-09-19 01:50:22 - 2015-10-26 18:49:01 3.770740e+01 days
## 2  2015-09-19 01:59:52 - 2015-10-26 17:50:36 3.766023e+01 days
## 3  2015-09-13 04:58:46 - 2015-09-13 05:01:49 2.116718e-03 days
## 4  2015-09-12 10:45:07 - 2015-09-12 10:52:20 5.009571e-03 days
## 5  2015-09-12 04:40:34 - 2015-09-12 04:41:29 6.350162e-04 days
## 6  2015-09-11 13:28:47 - 2015-09-11 13:31:07 1.622819e-03 days
## 7  2015-10-26 14:00:28 - 2015-10-26 15:06:15 4.567953e-02 days
## 8  2015-10-26 14:13:14 - 2015-10-26 14:35:13 1.526391e-02 days
## 9  2015-10-26 12:29:59 - 2015-10-26 12:37:16 5.050716e-03 days
## 10 2015-09-02 10:53:18 - 2015-09-02 10:57:21 2.822315e-03 days
## 11 2015-09-02 09:50:24 - 2015-09-02 09:54:22 2.751648e-03 days
## 12 2015-09-08 15:58:13 - 2015-09-08 16:02:41 3.104554e-03 days
## 13 2015-09-02 10:03:19 - 2015-09-02 10:06:09 1.975596e-03 days
## 14 2015-09-02 09:06:13 - 2015-09-02 09:14:27 5.715104e-03 days
## 15 2015-09-02 07:33:09 - 2015-09-02 07:44:13 7.691014e-03 days
## 16 2015-09-02 09:21:27 - 2015-09-02 09:21:39 1.411134e-04 days
## 17 2015-09-08 13:45:14 - 2015-09-08 13:46:08 6.350289e-04 days
## 18 2015-10-20 17:43:24 - 2015-10-20 17:46:58 2.469274e-03 days
## 19 2015-10-20 16:35:38 - 2015-10-20 16:42:25 4.713924e-03 days
## 20 2015-09-02 02:44:50 - 2015-09-02 04:03:54 5.490323e-02 days
## 21 2015-10-20 10:20:33 - 2015-10-20 10:22:20 1.234657e-03 days
## 22 2015-08-18 17:10:31 - 2015-10-20 02:24:52 6.238496e+01 days
## 23 2015-08-14 12:01:58 - 2015-09-07 21:09:06 2.437995e+01 days
## 24 2015-08-12 08:18:55 - 2015-09-02 14:41:39 2.126578e+01 days
## 25 2015-08-12 19:32:59 - 2015-10-25 18:17:05 7.394729e+01 days
## 26 2015-08-19 08:45:38 - 2015-10-20 15:05:47 6.226399e+01 days
## 27 2015-08-19 09:30:47 - 2015-09-14 17:47:31 2.634495e+01 days
## 28 2015-10-26 07:12:04 - 2015-10-26 07:30:49 1.301923e-02 days
## 29 2015-10-26 08:31:28 - 2015-10-26 08:41:58 7.295102e-03 days
## 30 2015-10-26 06:35:53 - 2015-10-26 06:59:29 1.638596e-02 days
## 31 2015-10-26 07:27:06 - 2015-10-26 07:29:31 1.683507e-03 days
## 32 2015-10-26 07:19:49 - 2015-10-26 07:22:44 2.020196e-03 days
```

```
mean <- mean(sum3$tot.ts) ## mean time
mean  # print mean
```

```
## Time difference of 10.81745 days
```

```
N <- length(sum3$tot.ts) ## number of observations
N    # print N
```

```
## [1] 32
```

```
sd <- sd(sum3$tot.ts) ## SD of time
sd  # print sd
```

```
## [1] 21.30664
```

```
se <- sd / sqrt(N) ## SE of time
se  # print se
```

```
## [1] 3.766517
```

Visualizing movement of tags

A quick and easy way to visualize the position of tags over time is to plot detections by latitude or longitude.

Figure 5. Plot of all detections by latitude, faceted by tag id.

```
p <- ggplot(subset(tags), aes(x = ts, y = lat))
p + geom_point(pch = 21) + theme_bw() + facet_wrap(~id, ncol = 4) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

## Warning: Removed 169 rows containing missing values (geom_point).
```

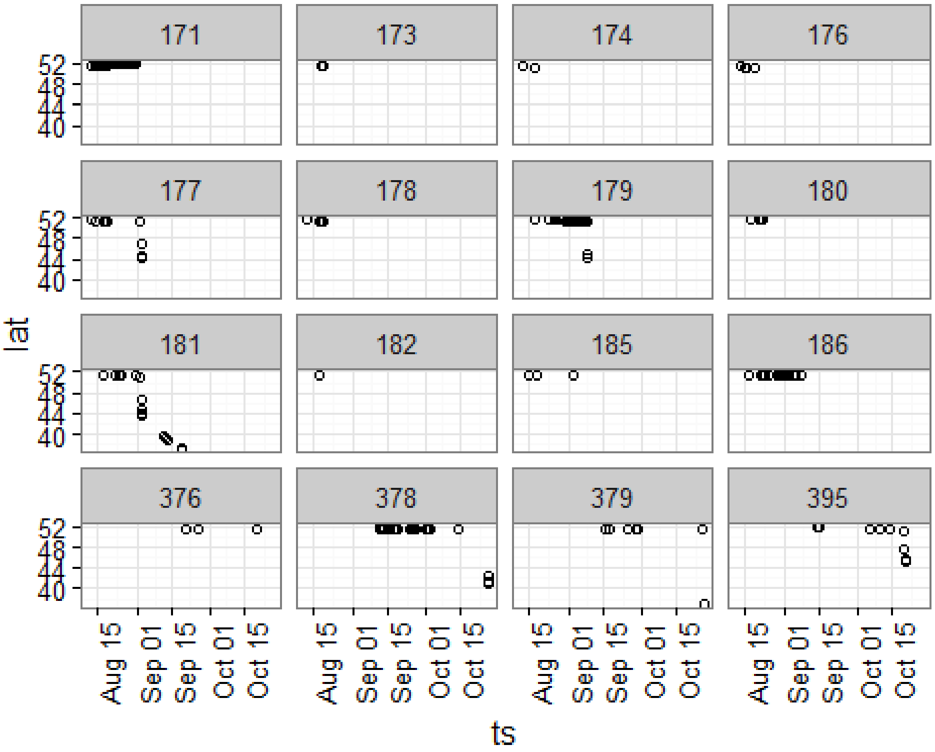


Figure 6. Plot of all detections by latitude, with colour representing tag id. Lines are added to visualize the tracks of individuals.

```
p <- ggplot(tags, aes(ts, lat, colour = id))
p + geom_point(pch = 21) + theme_bw() + geom_line()

## Warning: Removed 169 rows containing missing values (geom_point).
```

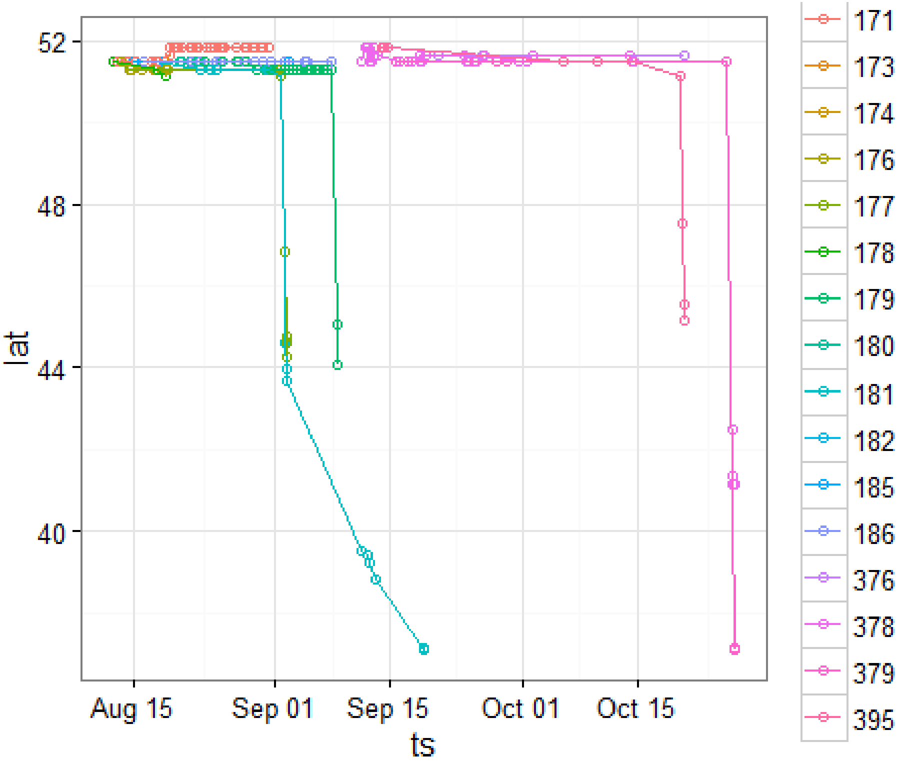


Figure 7. Plot of signal strength over time for tag 378 at the Piskwamish site. Colours represent signal strength at the different antenna.

```
p <- ggplot(subset(tags, id == "378" & site == "Piskwamish"), aes(ts, sig,
colour = ant))
p + geom_point(pch = 21) + theme_bw() +
  labs(title = "Tag 378 at Piskwamish") +
  xlab("Time") + ylab("Signal strength")## adds title to plot
```

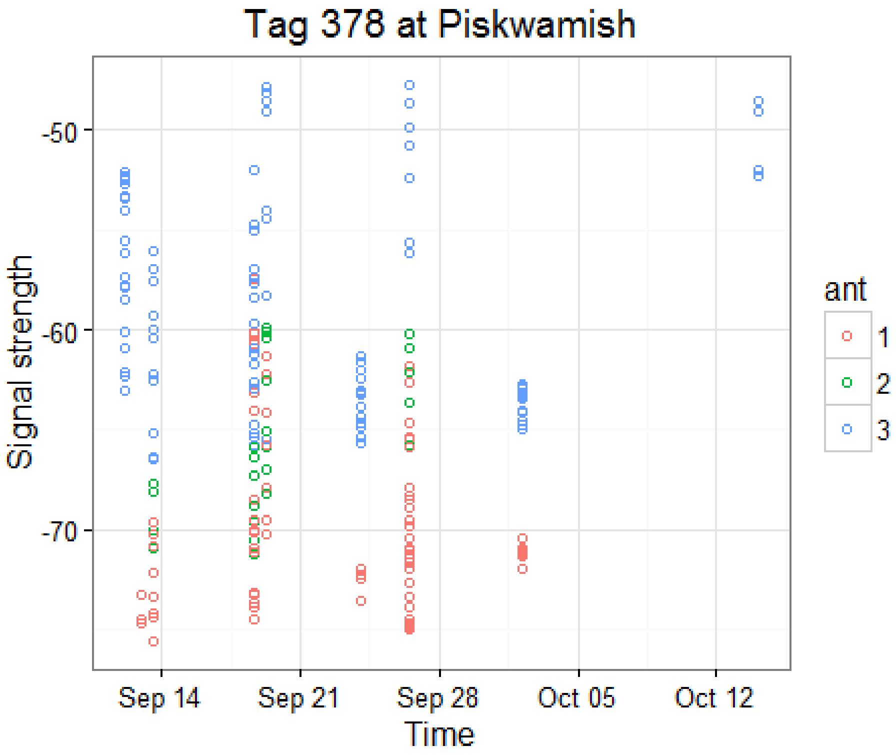


Figure 8. Plot of the change in latitude over time for tag 181.

```
p <- ggplot(subset(tags, id == "181"), aes(ts, lat))
p + geom_point(pch = 21) + theme_bw() +
  labs(title = "Tag 181") + ## adds title to plot
  theme(axis.title.x=element_blank(), axis.title.y=element_blank()) ##
removes x and y axis labels
```

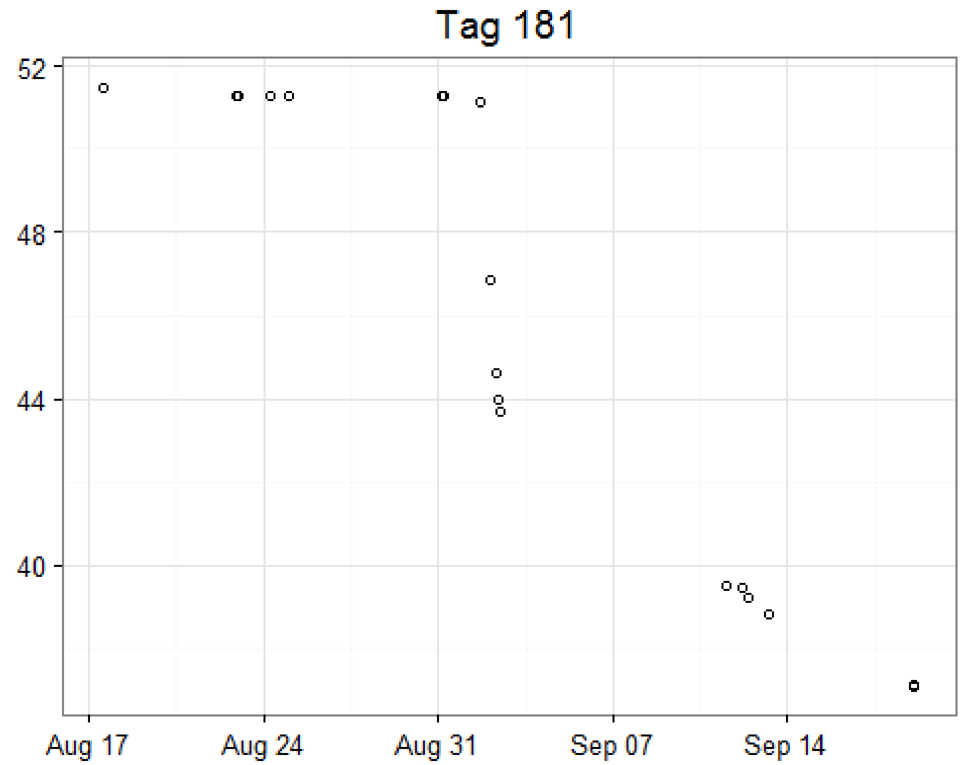


Figure 9. Plot of the movement of tag 181 among sites, where sites are ordered by latitude, with more northern sites at the top of the plot.

```
p <- ggplot(subset(tags, id == "181"), aes(ts, site))
p + geom_point(pch = 21) + theme_bw() +
  labs(title = "Tag 181") + ## adds title to plot
  theme(axis.title.x=element_blank(), axis.title.y=element_blank()) ##
removes x and y axis labels
```

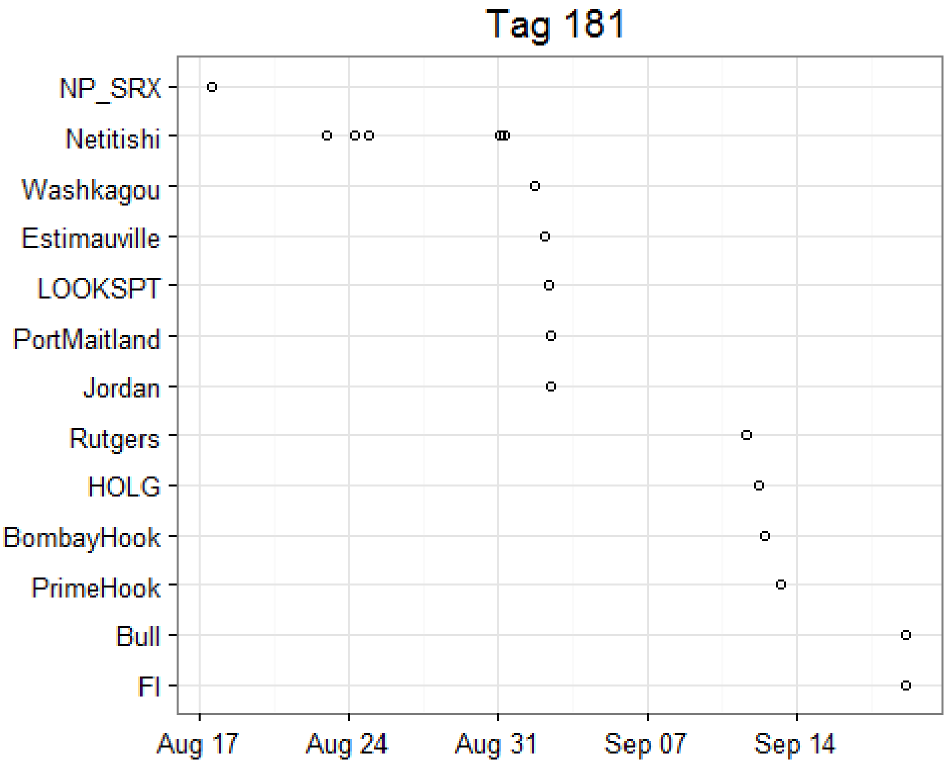


Figure 10. Signal strength for tag 181 during a specified time interval. Colour specifies the antenna, and shape specifies the site associated with the detections. In this case, tag 181 was detected by only one antenna during the specified time interval.

```
p <- ggplot(subset(tags, id == "181" & ts > as.POSIXct("2015-09-02 09:00") &
ts < as.POSIXct("2015-09-02 10:00")),
            aes(ts, sig, colour = ant, shape = site))
p + geom_point() + theme_bw()
```

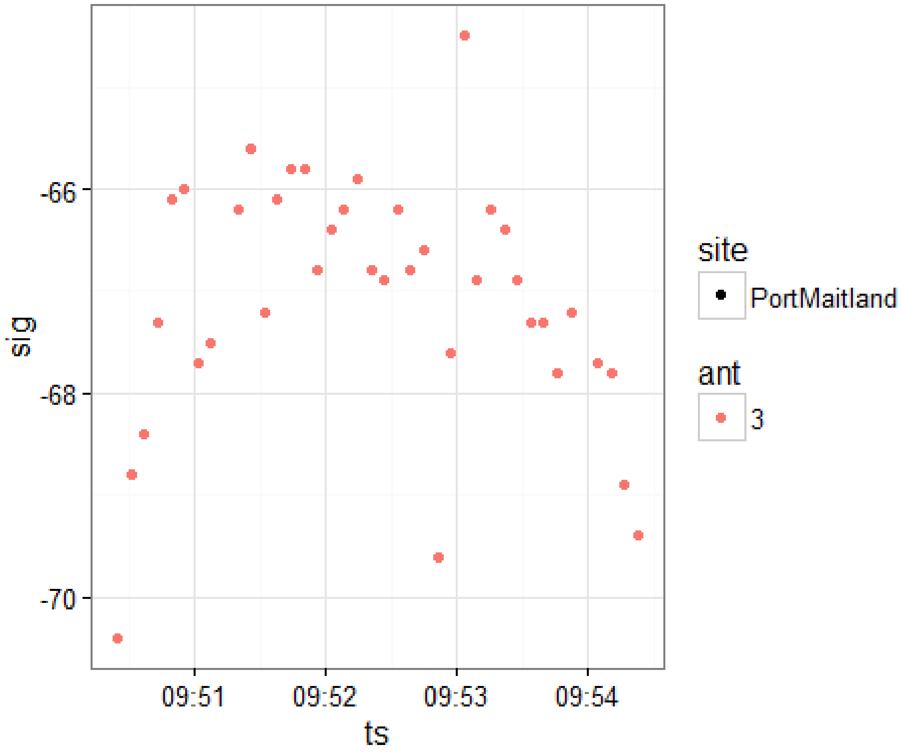


Figure 11. As with Figure 10, but over a different time interval, with colour specifying the antenna and shape specifying the site with detections. In this case the individual was detected by two antenna at the Lookspt site.

```
p <- ggplot(subset(tags, id == "181" & ts > as.POSIXct("2015-09-02 06:00") &
ts < as.POSIXct("2015-09-02 09:00")),
            aes(ts, sig, colour = ant, shape = site))
p + geom_point() + theme_bw()
```

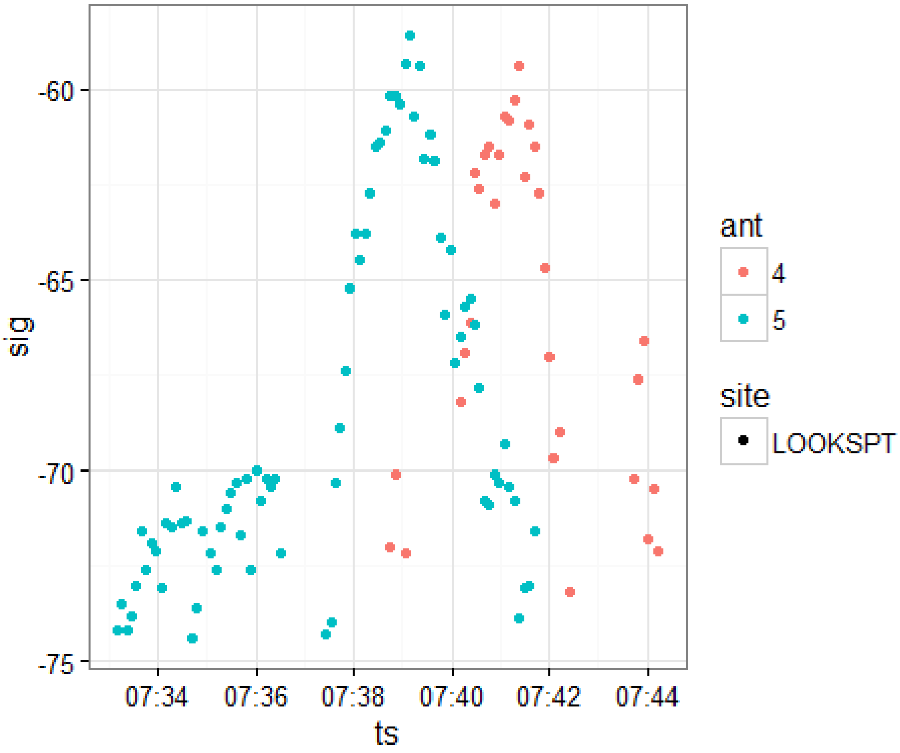
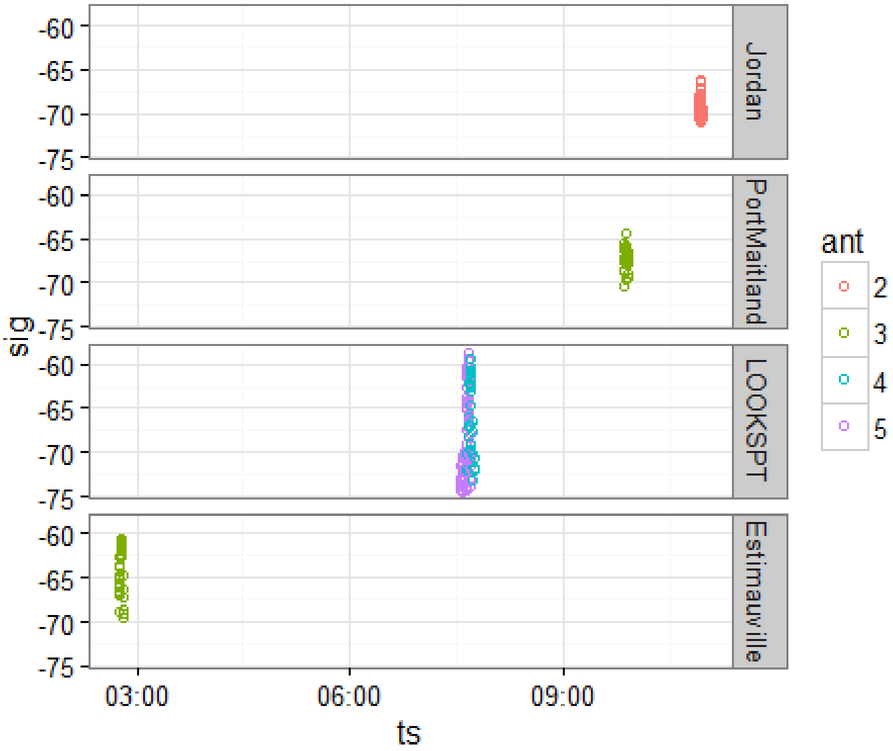


Figure 12. Signal strength of tag 181 during one night, faceted by site, and with colour representing antenna at each site.

```
p <- ggplot(subset(tags, id == "181" & ts > as.POSIXct("2015-09-02") & ts <
as.POSIXct("2015-09-03")), aes(ts, sig, colour = ant))
p + geom_point(pch = 21) + theme_bw() + facet_grid(site~.)
```



Map Detections

Create a base map to overlay movement tracks on:

```
gmap <-  get_map(location = c(lon = -72, lat = 45), ## lat/lon to center map
around
                 maptype = "hybrid", ## other options include "terrain",
"satellite", "roadmap"
                 source = "google",
                 zoom = 5) ## higher numbers zoom in more
```

Figure 13. Map all tag tracks, with colours regresenting tag ID.

```
p <- ggmap(gmap)
p + geom_path(data=tags, aes(lon, lat, group=fullID, col = fullID)) + ##
group will connect lines based on fullID
  labs(x = "Longitude", y = "Latitude", col = "ID") + theme_bw() +
  theme(legend.position = "none") ## removes legend
```

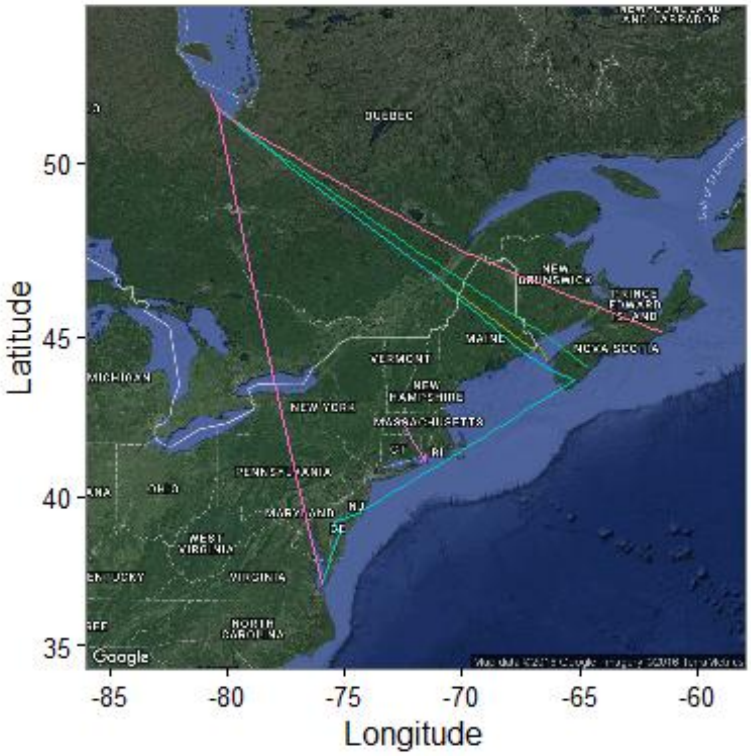
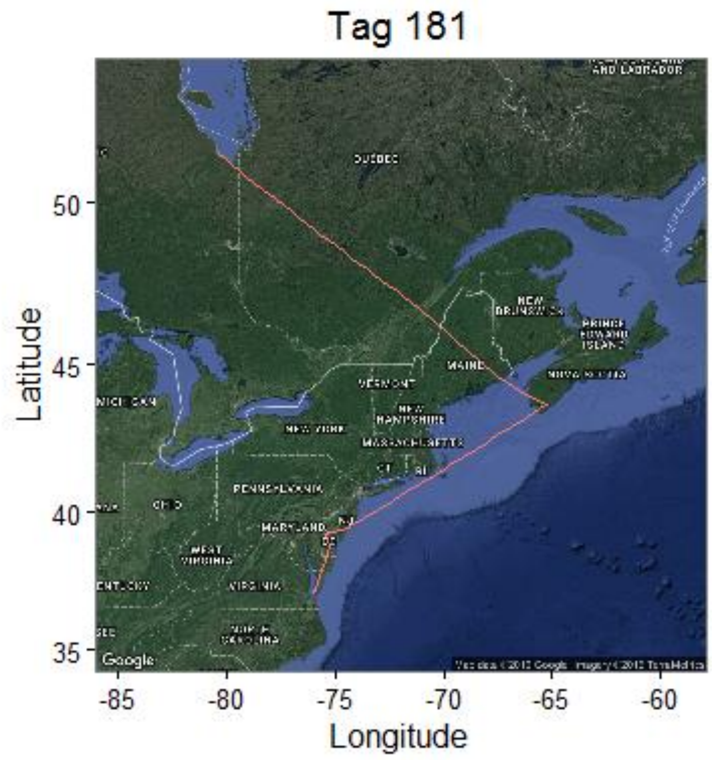


Figure 14. Map the track of tag ID 181.

```
p <- ggmap(gmap)
p + geom_path(data=subset(tags, id == "181"), aes(lon, lat,group=fullID, col
= fullID)) + ## group will connect lines based on fullID
  labs(title = "Tag 181", x = "Longitude", y = "Latitude", col = "ID") +
theme_bw() +
  theme(legend.position = "none") ## removes legend from plot
```


Tag 181
Latitude
50
45
40
35
-85
-80
-75
-70
-65
-60
Longitude
QUEBEC
NEW BRUNSWICK
PRINCE EDWARD ISLAND
MAINE
NOVA SCOTIA
VERMONT
NEW HAMPSHIRE
MASSACHUSETTS
NEW YORK
CT
RI
PENNSYLVANIA
OHIO
MARYLAND
NJ
DE
WEST VIRGINIA
VIRGINIA
NORTH CAROLINA
Google