translocation analysis - format data

steph

2024-09-04

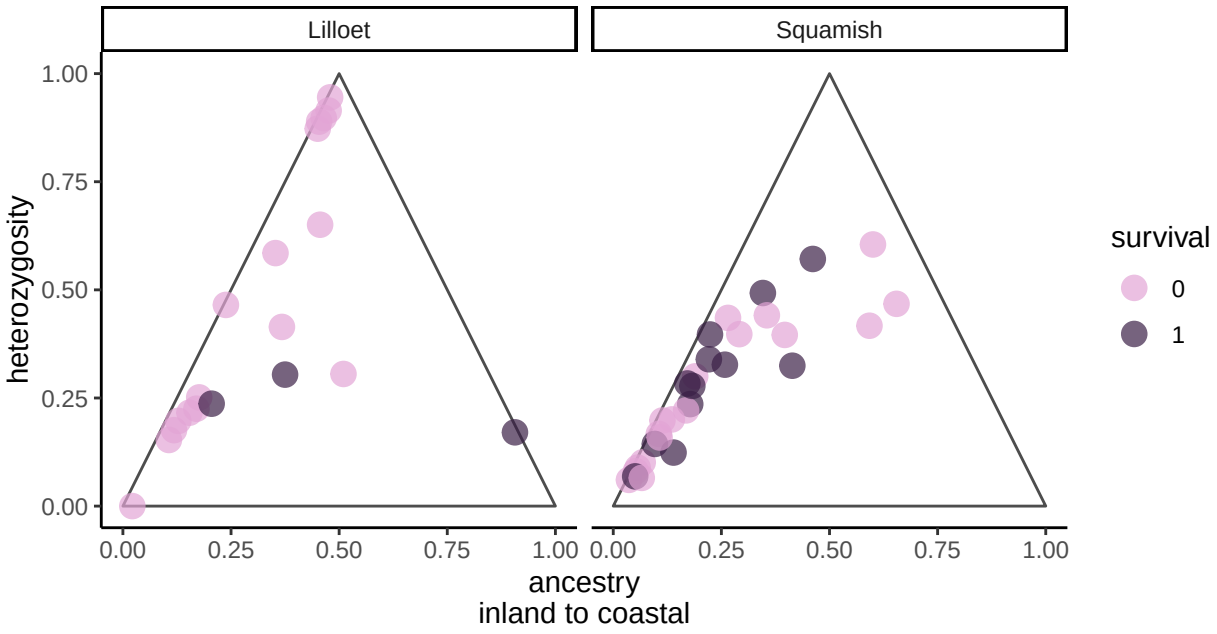
pre-process survival and phenotype data

load packages and read in initial dataset

- read in data
- create a merged metadata df

```
lat.df<-read_xlsx("C:/Users/Steph/TAMU_OneDrive/Thrushes/translocation/data_archiving/translocation_data
birds<-read_xlsx("C:/Users/Steph/TAMU_OneDrive/Thrushes/translocation/data_archiving/translocation_data
pembertonBirds<-read_xlsx("C:/Users/Steph/TAMU_OneDrive/Thrushes/translocation/data_archiving/translocat
meta_thrush<-rbind(birds,pembertonBirds)
```

```
poly.all<-data.frame(X.co=rep(c(0,0.5,1)), #make triangle polygon for plotting
                     Y.co=rep(c(0,1,0)))

ggplot(birds,aes(x=aims_ancestry,y=aims_heterozygosity,colour=as.character(survival)))+
  geom_polygon(data=poly.all,aes(x=X.co,y=Y.co),fill=NA,colour="grey30")+
  geom_point(size=4,shape=19,alpha=0.7)+
  scale_color_manual(values=natparks.pals("Arches2")[c(4,1)],name='survival')+
  coord_equal()+
  xlim(0,1)+ylim(0,1)+
  facet_wrap(vars(release_site))+
  xlab("ancestry\ninland to coastal")+
  ylab("heterozygosity")
```



format capture histories for input to CJS models

```
ch.df<-data.frame(matrix(NA,nrow=1,ncol=301))


#for each individual in translocated birds dataset
for(i in c(birds$name_in_vcf,pembertonBirds$name_in_vcf)){
  x1<-rep(0,300)
  x2<-rep(0,300)

  y<-lat.df%>%filter(name_in_vcf==i)%>%
    select(name_in_vcf,tagDays)%>%distinct()

  y1<-y%>%pull(tagDays)
  x1[c(1,y1[y1<301])]<-1 #add 1 to detections because bird was caught the day it was tagged
  if(length(y1)>0){if(max(y1)>300){
    x1[300]<-1  }} #fill in last day if bird caught later

  ch.df<-rbind(ch.df,c(i,x1))

}

colnames(ch.df)[1]<-"name_in_vcf"
if(is.na(ch.df[1,1])){ch.df<-ch.df[-1,]}

ch.df<-ch.df%>%unite(ch,-name_in_vcf,sep="")
```

```
ch.df.pemberton<-ch.df%>%filter(name_in_vcf%in%pembertonBirds$name_in_vcf)

ch.df<-ch.df%>%filter(name_in_vcf%in%birds$name_in_vcf)%>%
  left_join(birds)%>%
  mutate(ch=paste("days",ch,sep="_"))%>% #stops capture history from being saved as a very large number
  select(name_in_vcf,motustagid,aims_ancestry,release_site,sex_binary,ch)

#check that number of characters in detection history matches expectation for sets of 10 days
nchar(ch.df$ch[1])==(300+5)
```

```
## [1] TRUE
```

```
# write.csv(ch.df,
#           paste("C:/Users/Steph/GitHub/thrush_hybrids/translocation/thrush_survival_ch_",today,".csv"
#           )
```

- group detection histories into sets of ten days - 1 if detected in 10 day period

```
ch.df<-ch.df%>%mutate(ch2=gsub("days_","",ch))

ch2<-data.frame(name_in_vcf=ch.df$name_in_vcf)
days<-10
for(i in 1:round((nchar(ch.df$ch2)[1]/days))){
  colID=paste("t",i,sep="")
  ch2[colID]<-rep(0,nrow(ch.df))
  j=i*days-(days-1) #get starting position
  ch2[grep("1",substr(ch.df$ch2,j,j+(days-1))),colID]<-1
}

ch2<-ch2%>%unite(ch,-name_in_vcf,sep="")

ch2<-ch2%>%left_join(birds)%>%
  mutate(ch=paste("days",ch,sep="_"))%>% #stops capture history from being saved as a very large number
  select(name_in_vcf,motustagid,aims_ancestry,release_site,release_year,sex_binary,ch)
# write.csv(ch2,
#           paste("C:/Users/Steph/GitHub_data/translocation/survival.translocatedBirds.ch_10days",today
#           )


ch.df.pemberton<-ch.df.pemberton%>%mutate(ch2=gsub("days_","",ch))

ch2<-data.frame(name_in_vcf=ch.df.pemberton$name_in_vcf)
days<-10
for(i in 1:round((nchar(ch.df.pemberton$ch2)[1]/days))){
  colID=paste("t",i,sep="")
  ch2[colID]<-rep(0,nrow(ch.df.pemberton))
  j=i*days-(days-1) #get starting position
  ch2[grep("1",substr(ch.df.pemberton$ch2,j,j+(days-1))),colID]<-1
}


ch2<-ch2%>%unite(ch,-name_in_vcf,sep="")
```

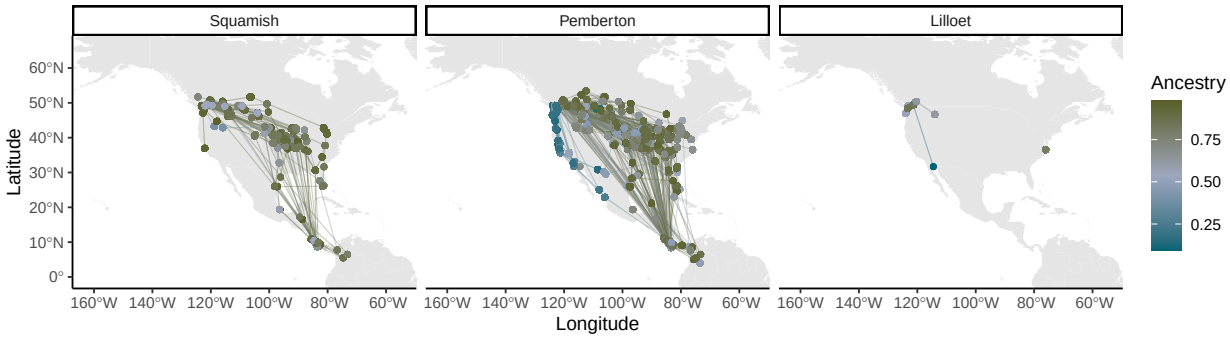
```
ch2<-ch2%>%left_join(pembertonBirds)%>%
  mutate(ch=paste("days",ch,sep="_"))%>% #stops capture history from being saved as a very large number
  select(name_in_vcf,motustagid,aims_ancestry,release_site,release_year,sex_binary,ch)

#check that number of characters in detection history matches expectation for sets of 10 days
nchar(ch2$ch[1])==(300/10+5)
```

```
## [1] TRUE
```

```
# write.csv(ch2,
#           paste("C:/Users/Steph/GitHub_data/translocation/survival.PembertonBirds.2020_2023.ch_10days
#           )
```

- make a map

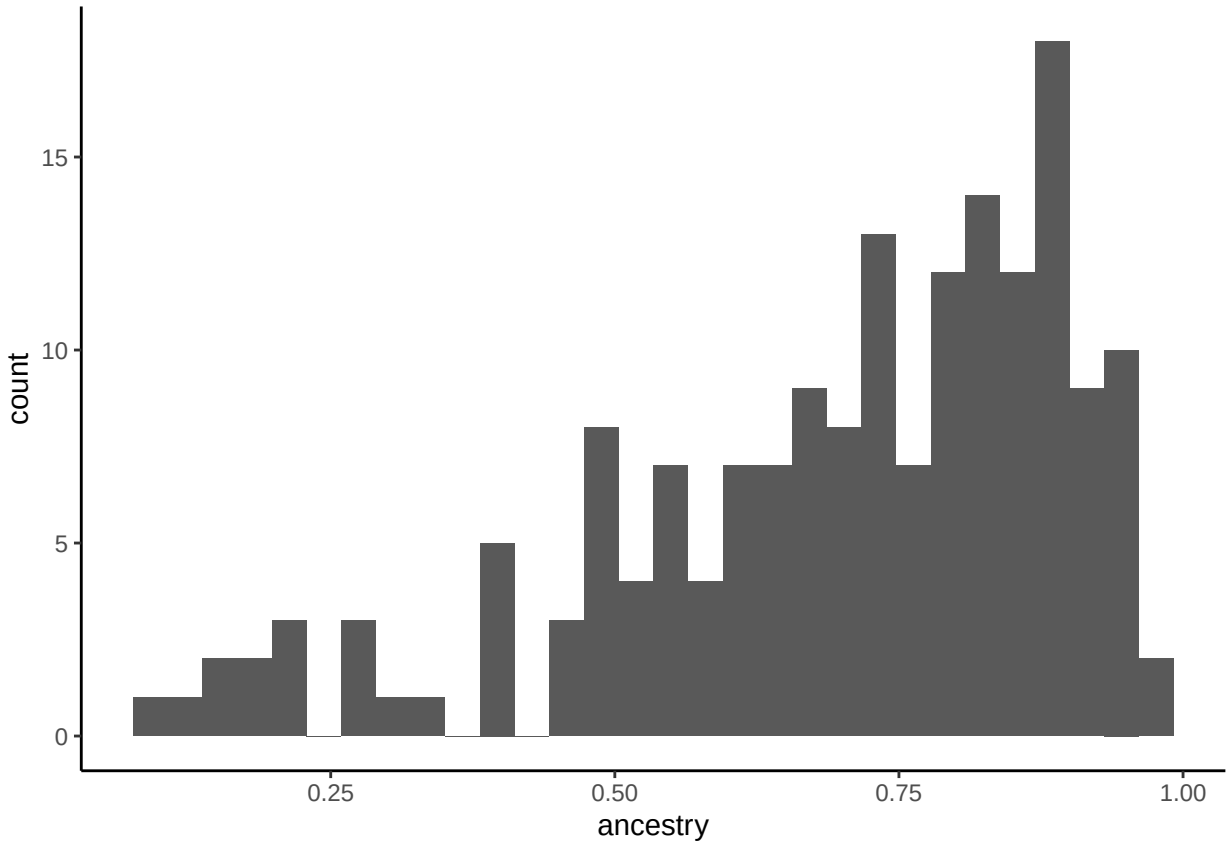
```
lat.df.birds<-lat.df%>%filter(name_in_vcf%in%c(birds$name_in_vcf,pembertonBirds$name_in_vcf))
#load map data
world <- ne_countries(scale = "medium", returnclass = "sf")

lat.df.birds$release_site<-factor(lat.df.birds$release_site,levels=c("Squamish","Pemberton","Lilloet"))
lat.df.birds<-lat.df.birds%>%left_join(meta_thrush%>%dplyr::select(name_in_vcf,aims_ancestry))
lat.df.birds$ancestry<-lat.df.birds$aims_ancestry*(-1)+1

map1<-ggplot() +
  geom_sf(data = world,fill="grey90",colour=NA)+
  geom_path(data=lat.df.birds,aes(x=recvDeployLon,y=recvDeployLat,
                                  group=name_in_vcf,colour=ancestry),size=0.3,alpha=0.4)+
  #scale_colour_manual(values=natparks.pals("Banff")[c(4,1)])+
  geom_point(data=lat.df.birds,aes(x=recvDeployLon,y=recvDeployLat,
                                   colour=ancestry,group=name_in_vcf),size=1.2)+
  #scale_shape_manual(values=c(19,15,17))+
  #scale_colour_gradientn(colours=c(natparks.pals("Banff")[4],"grey10",natparks.pals("Banff")[1]),
  #                       name="ancestry",values=c(0,0.5,1))+

  scale_colour_gradientn(colours=natparks.pals("Banff")[c(1,7,4)],
                         name="Ancestry",values=c(0,0.5,1))+
  xlab("Longitude")+ylab("Latitude")+
  #theme(legend.position="none")+
  ylim(0,66)+xlim(-162,-55)+
  facet_wrap(vars(release_site))

map1
```



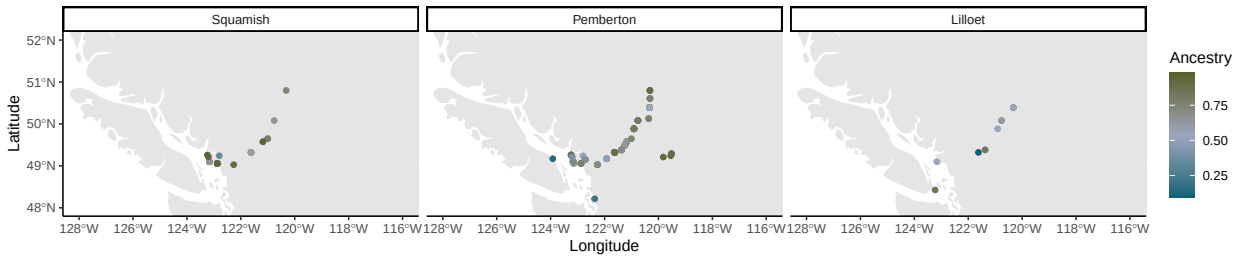

```
ggsave("C:/Users/Steph/TAMU_OneDrive/Thrushes/translocation/Fig4.png",
       width=10,height=3)
```

```
projection<-"+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"

lat.df.birds.fall1<-lat.df%>%
  filter(name_in_vcf%in%c(birds$name_in_vcf,pembertonBirds$name_in_vcf))%>%
  mutate(deployyear=as.numeric(substr(tagDeployStart,1,4)))%>%
  filter(detectyear==deployyear)%>%
  filter(detectdoy>212&tagProjID==280)%>%
  mutate(deployDay=substr(tagDeployStart,1,10),tsDay=substr(ts,1,10))%>%
  mutate(tagDays=as.integer(gsub(" days"," ",difftime(tsDay,deployDay,units="days"))))%>%
  select(motustagid,tsDay,tagDeployStart,tagDepLat,tagDepLon,
         recvDeployID,recvDeployLat,recvDeployLon,recvDeployName,
         detectyear,tagDays,detectdoy,deployDay,
         deployyear)%>%
  mutate(tagDeployStart=substr(tagDeployStart,1,10))%>%
  distinct%>%arrange(tagDays)%>%
  filter(!is.na(tagDays))%>%
  filter(motustagid!=85685)%>% #broken tag
  mutate(tagDepLat=if_else(motustagid%in%c(91383,85681),50.133366,tagDepLat),
         tagDepLon=if_else(motustagid%in%c(91383,85681),-122.95036,tagDepLon))

#estimate the distance between release site and detection site
#estimation needs to be pairwise, otherwise function will generate a matrix of all possible combos
lat.df.birds.fall1$releaseSiteDistance<-NA
for(i in 1:nrow(lat.df.birds.fall1)){
  lat.df.birds.fall1$releaseSiteDistance[i]<-
    distm(lat.df.birds.fall1[i,c("tagDepLon","tagDepLat")],
          lat.df.birds.fall1[i,c("recvDeployLon","recvDeployLat")],
          fun = distHaversine)
}


lat.df.birds.fall1<-lat.df.birds.fall1%>%
  filter(releaseSiteDistance<300000)%>%
    #keep the first record within the tier for each bird
    arrange(tagDays)%>%filter(!duplicated(motustagid))%>%
  mutate(motustagid=as.character(motustagid))%>%
  left_join(meta_thrush)%>%
  mutate(ancestry=aims_ancestry*(-1)+1)%>%
```

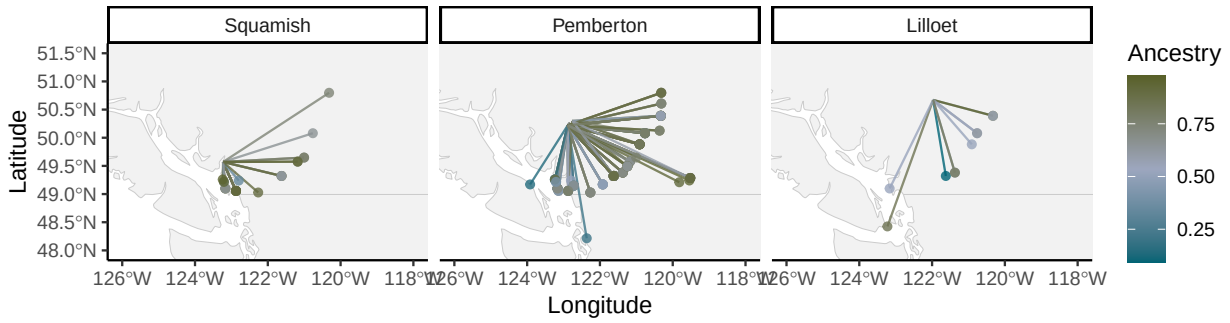
```
  mutate(release_site=factor(release_site,levels=c("Squamish","Pemberton","Lilloet")))

lat.df.birds.fall1.summary<-lat.df.birds.fall1%>%
  dplyr::select(ancestry,name_in_vcf,release_site,recvDeployLat,tagDepLat)%>%
  pivot_longer(c(tagDepLat,recvDeployLat),names_to='detect_type',values_to='latitude')%>%
  mutate(detect_type=gsub('Lat','',detect_type))%>%
  left_join(
    lat.df.birds.fall1%>%
      dplyr::select(name_in_vcf,recvDeployLon,tagDepLon)%>%
      pivot_longer(c(tagDepLon,recvDeployLon),names_to='detect_type',values_to='longitude')%>%
      mutate(detect_type=gsub('Lon','',detect_type)))


#add bearing measure - check that bearing estimate matches map
lat.df.birds.fall1.summary<-
  lat.df.birds.fall1.summary%>%
  left_join(rbind(birds%>%select(name_in_vcf,fall_releaseSite_bearing_tier1),
                  pembertonBirds%>%select(name_in_vcf,fall_releaseSite_bearing_tier1)))
lat.df.birds.fall1<-lat.df.birds.fall1%>%
  left_join(rbind(birds%>%select(name_in_vcf,fall_releaseSite_bearing_tier1),
                  pembertonBirds%>%select(name_in_vcf,fall_releaseSite_bearing_tier1)))

ggplot(lat.df.birds.fall1,aes(x=ancestry))+
  geom_histogram()
```



```
ggplot() +
  geom_sf(data = world,fill="grey90",colour=NA)+
  geom_path(data=lat.df.birds.fall1,aes(x=recvDeployLon,y=recvDeployLat,
                                        group=name_in_vcf,colour=ancestry),size=0.4,alpha=0.5)+
  #scale_colour_manual(values=natparks.pals("Banff")[c(4,1)])+
  geom_point(data=lat.df.birds.fall1,aes(x=recvDeployLon,y=recvDeployLat,
                                       colour=ancestry,group=name_in_vcf),size=1.5)+
  #scale_shape_manual(values=c(19,15,17))+
  #scale_colour_gradientn(colours=c(natparks.pals("Banff")[4],"grey10",natparks.pals("Banff")[1]),
  #                       name="ancestry",values=c(0,0.5,1))+

  scale_colour_gradientn(colours=natparks.pals("Banff")[c(1,7,4)],
                         name="Ancestry",values=c(0,0.5,1))+
  xlab("Longitude")+ylab("Latitude")+
  #theme(legend.position="none")+
  ylim(48,52)+xlim(-128,-116)+
  facet_wrap(vars(release_site))
```




```
ggplot() +
  geom_sf(data = world,fill="grey95",colour="grey80")+
  geom_path(data=lat.df.birds.fall1.summary,
            aes(x=longitude,y=latitude,
                group=name_in_vcf,colour=ancestry),size=0.5,alpha=0.8)+
  #scale_colour_manual(values=natparks.pals("Banff")[c(4,1)])+
  geom_point(data=lat.df.birds.fall1,aes(x=recvDeployLon,y=recvDeployLat,
                                       colour=ancestry,group=name_in_vcf),size=1.5,alpha=0.8)+
  #scale_shape_manual(values=c(19,15,17))+
  #scale_colour_gradientn(colours=c(natparks.pals("Banff")[4],"grey10",natparks.pals("Banff")[1]),
  #                       name="ancestry",values=c(0,0.5,1))+

  scale_colour_gradientn(colours=natparks.pals("Banff")[c(1,7,4)],
                         name="Ancestry",values=c(0,0.5,1))+
  xlab("Longitude")+ylab("Latitude")+
  #theme(legend.position="none")+
  ylim(48,51.5)+xlim(-126,-118)+
  facet_wrap(vars(release_site))
```




```
ggsave("C:/Users/Steph/TAMU_OneDrive/Thrushes/translocation/Fig3C.png",
       width=8,height=3)
```

verify that estimated bearings match where the birds are going

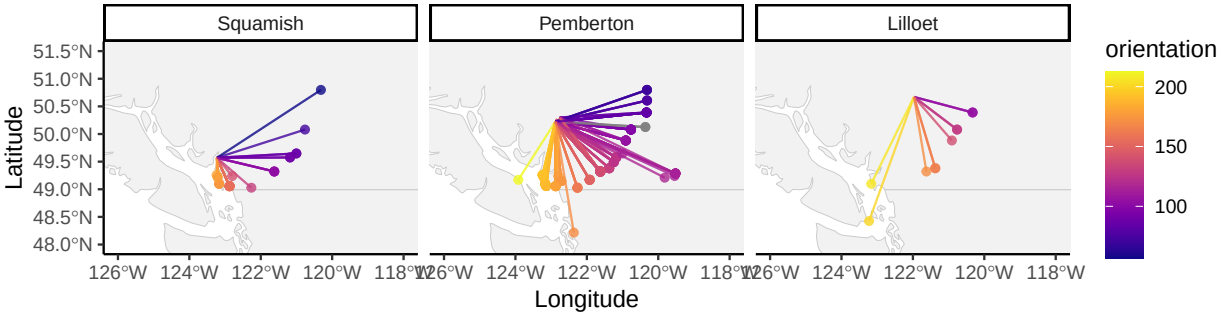
```
lat.df.birds.fall1.summary$fall_releaseSite_bearing_tier1
```

```
##   [1] "137.777831571093"   "137.777831571093"   "155.74914043148499"
##   [4] "155.74914043148499" "189.20395998029301" "189.20395998029301"
##   [7] "137.777831571093"   "137.777831571093"   "155.74914043148499"
##  [10] "155.74914043148499" "175.02440948559001" "175.02440948559001"
##  [13] "189.20395998029301" "189.20395998029301" "191.69297281222899"
##  [16] "191.69297281222899" "189.22921814863301" "189.22921814863301"
##  [19] "206.10415618863499" "206.10415618863499" "109.282966618739"
##  [22] "109.282966618739"   "143.06171849620401" "143.06171849620401"
##  [25] "95.905061698849707" "95.905061698849707" "155.74914043148499"
##  [28] "155.74914043148499" "155.74914043148499" "155.74914043148499"
##  [31] "161.341558305538"   "161.341558305538"   "143.06171849620401"
##  [34] "143.06171849620401" "189.20395998029301" "189.20395998029301"
##  [37] "95.874628065039701" "95.874628065039701" "104.883600191775"
##  [40] "104.883600191775"   "188.162375593788"   "188.162375593788"
##  [43] "137.777831571093"   "137.777831571093"   "193.37270790603799"
##  [46] "193.37270790603799" "193.37270790603799" "193.37270790603799"
##  [49] "129.51707069036101" "129.51707069036101" "193.37270790603799"
##  [52] "193.37270790603799" "179.67209947601799" "179.67209947601799"
##  [55] "143.06167047275699" "143.06167047275699" "174.25942416208099"
##  [58] "174.25942416208099" "84.082750664617294" "84.082750664617294"
##  [61] "86.903032628792303" "86.903032628792303" "175.02440948559001"
##  [64] "175.02440948559001" "130.71112059247201" "130.71112059247201"
##  [67] "177.5718196334"     "177.5718196334"     "149.27856196954801"
##  [70] "149.27856196954801" "137.777831571093"   "137.777831571093"
##  [73] "130.70391910296601" "130.70391910296601" "137.788248313694"
##  [76] "137.788248313694"   "139.64101315245301" "139.64101315245301"
##  [79] "130.72664985437601" "130.72664985437601" "179.67209947601799"
##  [82] "179.67209947601799" "NA"                 "NA"
##  [85] "189.22921814863301" "189.22921814863301" "139.26341829446099"
##  [88] "139.26341829446099" "105.089857112007"   "105.089857112007"
```

```
##  [91] "176.736191355062"   "176.736191355062"   "137.788248313694"
##  [94] "137.788248313694"   "149.27856196954801" "149.27856196954801"
##  [97] "120.417227278659"   "120.417227278659"   "137.777831571093"
## [100] "137.777831571093"   "120.417227278659"   "120.417227278659"
## [103] "84.091125355635299" "84.091125355635299" "175.02440948559001"
## [106] "175.02440948559001" "189.22921814863301" "189.22921814863301"
## [109] "161.359252060871"   "161.359252060871"   "84.082750664617294"
## [112] "84.082750664617294" "120.42047356006201" "120.42047356006201"
## [115] "84.091125355635299" "84.091125355635299" "191.66476415631601"
## [118] "191.66476415631601" "188.162375593788"   "188.162375593788"
## [121] "130.70391910296601" "130.70391910296601" "143.06171849620401"
## [124] "143.06171849620401" "122.371416937552"   "122.371416937552"
## [127] "70.529780966839297" "70.529780966839297" "139.54392570516899"
## [130] "139.54392570516899" "188.18673634714199" "188.18673634714199"
## [133] "183.39080102452101" "183.39080102452101" "103.378429822249"
## [136] "103.378429822249"   "103.378429822249"   "103.378429822249"
## [139] "86.903032628792303" "86.903032628792303" "113.25546344579899"
## [142] "113.25546344579899" "189.22921814863301" "189.22921814863301"
## [145] "104.880721487504"   "104.880721487504"   "117.016740233527"
## [148] "117.016740233527"   "95.868589062385198" "95.868589062385198"
## [151] "113.25546344579899" "113.25546344579899" "143.06167047275699"
## [154] "143.06167047275699" "115.82474613705099" "115.82474613705099"
## [157] "163.43506512962099" "163.43506512962099" "130.70391910296601"
## [160] "130.70391910296601" "125.42161452451001" "125.42161452451001"
## [163] "137.777831571093"   "137.777831571093"   "137.777831571093"
## [166] "137.777831571093"   "95.874628065039701" "95.874628065039701"
## [169] "137.777831571093"   "137.777831571093"   "86.664631172734602"
## [172] "86.664631172734602" "139.54392570516899" "139.54392570516899"
## [175] "124.51991573872399" "124.51991573872399" "NA"
## [178] "NA"                 "181.096776021133"   "181.096776021133"
## [181] "130.71112059247201" "130.71112059247201" "179.67209947601799"
## [184] "179.67209947601799" "84.082750664617294" "84.082750664617294"
## [187] "137.788248313694"   "137.788248313694"   "124.51991573872399"
## [190] "124.51991573872399" "113.25546344579899" "113.25546344579899"
## [193] "113.25546344579899" "113.25546344579899" "95.874628065039701"
## [196] "95.874628065039701" "105.089857112007"   "105.089857112007"
## [199] "127.478388261454"   "127.478388261454"   "95.874628065039701"
## [202] "95.874628065039701" "174.22640148903"    "174.22640148903"
## [205] "95.874628065039701" "95.874628065039701" "104.883600191775"
## [208] "104.883600191775"   "161.341558305538"   "161.341558305538"
## [211] "155.74914043148499" "155.74914043148499" "137.788248313694"
## [214] "137.788248313694"   "70.518648514139798" "70.518648514139798"
## [217] "149.29307585835099" "149.29307585835099" "115.296963352897"
## [220] "115.296963352897"   "84.082750664617294" "84.082750664617294"
## [223] "76.719621586223496" "76.719621586223496" "113.25546344579899"
## [226] "113.25546344579899" "149.29307585835099" "149.29307585835099"
## [229] "76.729649863680905" "76.729649863680905" "143.06171849620401"
## [232] "143.06171849620401" "155.74914043148499" "155.74914043148499"
## [235] "155.74914043148499" "155.74914043148499" "103.378429822249"
## [238] "103.378429822249"   "212.51434355391601" "212.51434355391601"
## [241] "193.37270790603799" "193.37270790603799" "137.777831571093"
## [244] "137.777831571093"   "137.777831571093"   "137.777831571093"
## [247] "170.369678540443"   "170.369678540443"   "155.74914043148499"
## [250] "155.74914043148499" "114.386507516969"   "114.386507516969"
```

```
## [253] "89.761649716884705" "89.761649716884705" "86.646131360719195"
## [256] "86.646131360719195" "104.883600191775"   "104.883600191775"
## [259] "174.25942416208099" "174.25942416208099" "179.648723877099"
## [262] "179.648723877099"   "161.341558305538"   "161.341558305538"
## [265] "179.67209947601799" "179.67209947601799" "124.51991573872399"
## [268] "124.51991573872399" "104.880721487504"   "104.880721487504"
## [271] "89.761649716884705" "89.761649716884705" "189.22921814863301"
## [274] "189.22921814863301" "174.25942416208099" "174.25942416208099"
## [277] "137.777831571093"   "137.777831571093"   "113.25546344579899"
## [280] "113.25546344579899" "137.788248313694"   "137.788248313694"
## [283] "70.518648514139798" "70.518648514139798" "137.788248313694"
## [286] "137.788248313694"   "127.478388261454"   "127.478388261454"
## [289] "149.27856196954801" "149.27856196954801" "84.082750664617294"
## [292] "84.082750664617294" "72.210154579150398" "72.210154579150398"
## [295] "56.636805160889899" "56.636805160889899" "177.5718196334"
## [298] "177.5718196334"     "193.37270790603799" "193.37270790603799"
## [301] "149.29307585835099" "149.29307585835099" "143.12300522071101"
## [304] "143.12300522071101" "170.58444073283999" "170.58444073283999"
## [307] "99.665910255320597" "99.665910255320597" "76.729649863680905"
## [310] "76.729649863680905" "191.69297281222899" "191.69297281222899"
## [313] "139.54392570516899" "139.54392570516899" "163.43506512962099"
## [316] "163.43506512962099" "199.96064339232399" "199.96064339232399"
## [319] "174.25942416208099" "174.25942416208099" "137.777831571093"
## [322] "137.777831571093"   "109.270195529365"   "109.270195529365"
## [325] "137.777831571093"   "137.777831571093"   "84.091125355635299"
## [328] "84.091125355635299" "174.22640148903"    "174.22640148903"
## [331] "142.98028635803601" "142.98028635803601" "124.517490084063"
## [334] "124.517490084063"   "137.777831571093"   "137.777831571093"
## [337] "104.880721487504"   "104.880721487504"   "137.788248313694"
## [340] "137.788248313694"   "137.788248313694"   "137.788248313694"
## [343] "179.648723877099"   "179.648723877099"   "137.788248313694"
## [346] "137.788248313694"
```

```
ggplot() +
  geom_sf(data = world,fill="grey95",colour="grey80")+
  geom_path(data=lat.df.birds.fall1.summary,
            aes(x=longitude,y=latitude,
                group=name_in_vcf,colour=as.numeric(fall_releaseSite_bearing_tier1)),
            size=0.5,alpha=0.8)+
  geom_point(data=lat.df.birds.fall1,aes(x=recvDeployLon,y=recvDeployLat,
                                colour=as.numeric(fall_releaseSite_bearing_tier1),
                                group=name_in_vcf),size=1.5,alpha=0.8)+

  scale_colour_viridis(option='plasma',name="orientation")+
  xlab("Longitude")+ylab("Latitude")+
  ylim(48,51.5)+xlim(-126,-118)+
  facet_wrap(vars(release_site))
```


Squamish
Pemberton
Lilloet
orientation
200
150
100
Latitude
51.5°N
51.0°N
50.5°N
50.0°N
49.5°N
49.0°N
48.5°N
48.0°N
126°W
124°W
122°W
120°W
118°W
Longitude