

# Proposed individuals for sequencing

This is a list of possible individuals for the next round of sequencing and their respective numbers. Each have their own costs and benefits which I will go into in some of the sections. I would appreciate feedback on which sets of individuals you think would be worth sequencing. I have included the R code I used to identify these individuals, so feel free to take a look if you are interested, just click the ‘show’ button (I’m always happy to have feedback on my code!)

## Groups of individuals:

1. Confirmed dead ids that died within 90 days of hatching (n=224)
2. Missing (almost certainly) dead ids (n=119)
3. Un-ringed individuals with known death dates (n=73)
4. Siblings of confirmed dead individuals in 1 (n=78)
5. Siblings of ids that went missing in 2 (n=116)
6. Siblings of previously sequenced ids that had high inbreeding coefficients (n=75)
7. Parents of individuals sequenced last time (n=67)
8. Parents of individuals proposed in 1 and 2 (n=25)
9. Siblings and parents of individuals proposed in 3 (n=62)

.... All of which have good blood samples and are born after 2010 as this is when regular visits started so we can be more sure of their death dates. This will also reduce variation generated between years, but will mean we won’t have temporal data if we want to look at allele freq change over time (sorry Tristan). *NOTE:* I couldn’t find any information on location of blood samples post 2020 so currently this list does not include these years. But it might be good to have?

- For inbreeding depression analysis depending on budgeting I would propose the priority individuals should be those from groups 1, 2 and 6 so a total of 418 unique individuals. (Group 2 could always be split into cross fostered and non-cross fostered).
- I’m worried that samples from individuals in 3 may be hard to find, but if they’re not then this would bump the total up to 491.
- If there is the possibility we may also want to look at h2, parental effects or family inheritance etc. later then I would suggest also sequencing 4, 5, 7 and 8. Total of another 382 unique individuals (3 ids are in the previous suggestion)

See below for details of individuals in each section and code and at the bottom a fig per year:

~ 1 ~

## Confirmed dead individuals (n=224)

- Known death date so in most cases the body was found or sometimes just the ring
- Known birth date (i.e. those born in the study area)
- Have a ring id

```
## getting individuals with known death and birth dates
age_dead_exact=age_dead_exact%>%
  filter(!is.na(HatchDate) & !is.na(DeathDate) & DeathDate != "") %>%
  mutate(
    hatch_date = as.Date(HatchDate, format = "%d/%m/%Y"), #mutating dates to correct formatting
    death_date = as.Date(DeathDate, format = "%d/%m/%Y"),
    age_at_death = death_date - hatch_date)

dead_wRing_knownDd_blood=age_dead_exact%>%
  filter(age_at_death<=90, #died before 90 days
    RingId != "", # has ring id
    hatch_date > as.Date("2010-01-01"))%>% # hatched after / in 2010
  anti_join(sequenced_ids, by = "RingId")%>% # remove ids sequenced last time
  left_join(lab, by = "RingId")%>% #joining with lab info
  filter(SampleType%in%c("blood", "Blood cell"))%>% #check which ones have blood samples
  distinct(RingId, .keep_all = TRUE) #
```

Number of ids with blood samples:

```
n_distinct(dead_wRing_knownDd_blood$RingId)
```

```
## [1] 224
```

Number of ids with DNA already extracted:

```
DNA_extr_1=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_Placement)
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(dead_wRing_knownDd_blood, by = 'RingId')%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_1$RingId)
```

```
## [1] 220
```

*Benefits* - Known dead so good for survival analysis

~ 2 ~

**Individuals that went ‘missing’ before they were 45 days old (n = 119)**

- When the last time they were observed they were <35 days old and others in the clutch have recorded measurements after. (probably a very conservative approach but better safe than sorry)
- These individuals do **not** have a recorded death date (for *reasons*) but they would be too young to have fledged so most likely died and the body fell far from the nest and was not recovered.
- Only for ids with blood.

See code block below as an example of how I flagged these individuals. I then manually verified these ids on the database using the ‘possibly dead’ box in the database and the timeline when needed. (see calc\_missing.R for full run through in how these ids were identified)

```

# Using the all the records from ID measurements and combining with info on clutch to group by the clutch
missings <- all_records_clutch %>%
  filter(ObservationDate <= as.Date(paste(Season+1, 1, 1, sep = "-"))) %>% ## filter for only observations from the next year
  mutate(age_at_latest=as.integer(LatestBirdObservationDate - HatchDate))%>%
  group_by(BornClutchId) %>%
  ## Checking if there's any observations within the same clutch later than the last obs of the individual
  # conditional on whether the observation date is within 35 days of the latest bird obs (in case observation date is missing)
  mutate(obs_after = as.integer(
    sapply(LatestBirdObservationDate, function(x) any((ObservationDate > x)& (ObservationDate <= x + 35)))) %>%
  ## define missing ids if the last obs is before other obs taken in the clutch, the chick is within 35 days of the latest obs
  # so couldnt possibly have fledged but records stop for that id (i.e. likely because it is dead)
  mutate(missing = case_when(
    obs_after==1 & (LatestBirdObservationDate<HatchDate+45) & (age_at_latest<=35) ~1
  ))%>%
  ungroup()%>%
  filter(missing==1)

```

Missing ids:

```

## missings_with_blood was expoted and checked manually with the database
# reading confirmed individuals back in.
missings=read.csv("Barn_owl_general/missing_ids/Confirmed_missingIDs_with_blood_and_rings.csv")%>%
  anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced (n=1)
  anti_join(dead_wRing_knownDd_blood, by = 'RingId')

n_distinct(missings$RingId)

```

```
## [1] 54
```

And missing cross fostered ids too:

```

##checked manually with the database
# reading confirmed individuals back in.
cfs_missings=read.csv("Barn_owl_general/missing_ids/Confirmed_cross_fost_missingIDs_with_blood.csv")%>%
  anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced (n=1)
  anti_join(dead_wRing_knownDd_blood, by = 'RingId')

n_distinct(cfs_missings$RingId)

```

```
## [1] 65
```

Total missing ids:

```

total_missings=rbind(cfs_missings,missings)%>%
  unique()

n_distinct(total_missings$RingId)

```

```
## [1] 119
```

Number of these that have had DNA extracted already:

```
DNA_extr_2=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_Placement)
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(total_missings, by = 'RingId')%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_2$RingId)
```

```
## [1] 118
```

*Benefits* - Can be pretty confident that these individuals are dead so will add to survival analysis

*Costs* - Cross fostered individuals might make it harder to be sure death is due to genetics vs being cross fostered

~ 3 ~

**All individuals born after 2010 with no RingId (n = 73)**

- Ids that died before they were old enough to be ringed
- At the moment I have only used those with a confirmed death date

.... But the problem is, how many of them have blood samples and where are they

**NOTE: We could get 139 more samples if we also assume that since these ids dont have a recorded RingID they probably died.**

(PLEASE CHECK: is there any way that these individuals could have been ringed at a later date but this was somehow not linked properly in the database? I dont believe so but just to be sure please check code)

e.g. first 10 ids:

```
All_after2010_noRing=bird%>%
  filter(as.Date(HatchDate, format = "%d/%m/%Y") > as.Date("2010-01-01"))%>% # hatched after / in 2010
  filter(RingId=="")

All_after2010_noRing[1:10, c(1:3,6,8,17,18)]
```

##	BirdId	RingId	Mark	DeathDate	BornClutchId	EarliestBirdObservationDate
## 1	9530		NB20101		1144	22/05/2010 00:00:09
## 2	9531		NB201010		1137	04/05/2010 23:00:00
## 3	9532		NB201011		1137	04/05/2010 23:00:00
## 4	9533		NB201012		1120	08/06/2010 00:00:09
## 5	9534		NB201013		1152	23/06/2010 22:00:00
## 6	9535		NB201014		1152	23/06/2010 22:00:00
## 7	9536		NB201015		1153	02/06/2010 00:00:00
## 8	9537		NB201016		1153	02/06/2010 00:00:00
## 9	9538		NB201017		1142	24/05/2010 00:00:14
## 10	9539		NB201018		1142	24/05/2010 00:00:14

```
## LatestBirdObservationDate
## 1 22/05/2010 10:00:00
## 2 04/05/2010 23:00:00
## 3 04/05/2010 23:00:00
## 4 08/06/2010 10:00:00
## 5 29/06/2010 20:00:00
## 6 29/06/2010 20:00:00
## 7 02/06/2010 00:00:00
## 8 02/06/2010 00:00:00
## 9 24/05/2010 14:00:00
## 10 24/05/2010 14:00:00
```

Can see that for a lot of them the first observation date was also the last observation. And also see these individuals have no RingId but they do have a mark starting with 'NB', which we will use to link to the lab info

Total number of these unringed ids:

```
n_distinct(All_after2010_noRing$BirdId)
```

```
## [1] 601
```

And number with known death dates:

```
clutch_born=clutch%>%
  rename(BornClutchId=Id)
# clutch_raised=clutch%>%
#   rename(RaisedClutchId=Id)

dead_noRing=All_after2010_noRing%>%
  filter(!(DeathDate==""))%>%
  left_join(clutch_born, by = 'BornClutchId')%>%
  select(BirdId, Mark, BornClutchId, Season, Nestbox.y, CrossFostered, HatchDate, DeathDate)

n_distinct(dead_noRing$BirdId)
```

```
## [1] 203
```

Now to link the ids to lab info by first filtering all the lab samples that start with 'NB' and then only keeping those that have blood samples.

Brief example showing 2 ids, one that has 2 sample names (one with a ring id and one in the NB format), and another that has only samples names in the NB format:

```
big_boy_lab[c(25056,27858),c(3:5, 14) ]
```

```
## Sample_Name_1 Sample_Name_2 Sample_Name_3 Tissue.Type
## 25056 M043254 NB438RB Blood cell
## 27858 NB42LB NB42LB Swab
```

Total number of individual samples with a sample name that starts with NB **AND** that doesn't have another sample name starting with an 'M' (i.e. individuals that did eventually get a ring) **AND** that have blood info:

```
## all lab info with samples that start with either M or NB
NBs=big_boy_lab%>%
  filter(str_starts(Sample_Name_1, 'NB') | str_starts(Sample_Name_2, 'NB'))%>%
  filter(!str_starts(Sample_Name_2, 'M') & !str_starts(Sample_Name_3, 'M') & !str_starts(Sample_Name_1, 'M'))%>%
  filter(Tissue.Type%in%c("blood", "Blood"))#check which ones have blood samples

n_distinct(NBs)
```

```
## [1] 582
```

So combining these two datasets we can search for ids with no Ring ID that match the format specified in the sample names in the lab info and attempted to match these with either sample name 1 or sample name 2 Formats include:

- NB, season, N, nestbox, mark
- NB, nestbox, Mark

```
# First sample same seq: NB, Season, N, nestbox, Mark
# and match to sample name 2

dead_noRing_knownDd_NB_S_N_nest_M=dead_noRing%>%
  drop_na(BornClutchId)%>% #remove 2 ids with no clutch info
  mutate(nestbox_edit=gsub("[A-Za-z]", "", Nestbox.y))%>%
  mutate(Sample_Name_2 = paste0('NB', Season, 'N', nestbox_edit, Mark))

sample_name2_match_NB_S_N_nest_M=dead_noRing_knownDd_NB_S_N_nest_M%>%
  inner_join(NBs, by = 'Sample_Name_2', relationship = "many-to-many")

#n_distinct(sample_name2_match_NB_S_N_nest_M$BirdId)

# NB, nestbox, Mark
# and match to sample name 1 & 2

dead_noRing_knownDd_NB_nest_mark=dead_noRing%>%
  drop_na(BornClutchId)%>% #remove 2 ids with no clutch info
  mutate(nestbox_edit=gsub("[A-Za-z]", "", Nestbox.y))%>%
  mutate(Sample_Name_2 = paste0('NB', nestbox_edit, Mark))

sample_name2_match_NB_nest_mark=dead_noRing_knownDd_NB_nest_mark%>%
  inner_join(NBs, by = 'Sample_Name_2', relationship = "many-to-many")

#n_distinct(sample_name2_match_NB_nest_mark$BirdId)

## sample name 1

dead_noRing_knownDd_NB_nest_mark=dead_noRing%>%
  drop_na(BornClutchId)%>% #remove 2 ids with no clutch info
  mutate(nestbox_edit=gsub("[A-Za-z]", "", Nestbox.y))%>%
  mutate(Sample_Name_1 = paste0('NB', nestbox_edit, Mark))

sample_name1_match_NB_nest_mark=dead_noRing_knownDd_NB_nest_mark%>%
  inner_join(NBs, by = 'Sample_Name_1', relationship = "many-to-many")
```

```
#n_distinct(sample_name1_match_NB_nest_mark$BirdId)

# N, nestbox, Mark
# and match to sample name 1
dead_noRing_knownDd_N_nest_M=dead_noRing%>%
  filter(CrossFostered==0 | is.na(CrossFostered))%>%
  drop_na(BornClutchId)%>% #remove 2 ids with no clutch info
  mutate(nestbox_edit=gsub("[A-Za-z]", "", Nestbox.y))%>%
  mutate(Sample_Name_1 = paste0('N', nestbox_edit, Mark))

sample_name1_match_N_nest_M=dead_noRing_knownDd_N_nest_M%>%
  inner_join(NBs, by = 'Sample_Name_1',relationship = "many-to-many")

#n_distinct(sample_name1_match_N_nest_M$BirdId)
```

Total number of unringed Ids with blood

```
all_unringed_linked_blood=rbind(sample_name2_match_NB_S_N_nest_M,sample_name1_match_NB_nest_mark)%>%
  rbind(sample_name2_match_NB_nest_mark)%>%
  rbind(sample_name1_match_N_nest_M)%>%
  unique()

n_distinct(all_unringed_linked_blood$BirdId)
```

```
## [1] 73
```

Total of these with DNA already extracted:

```
DNA_extr_3.1=all_unringed_linked_blood%>%
  select(Sample_Name_1, Sample_Name_2,BirdId,Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Bo
  filter(Date.extraction!='')

n_distinct(DNA_extr_3.1$BirdId)
```

```
## [1] 55
```

*Benefits* - Gives an estimator of (very) early life survival

*Costs* - Might be difficult to find these individuals in the freezers given they dont have a ring id. - May also be hard to keep track of them without a ring id.

~ 4 ~

Siblings of the ids that died from 1 (n=78)

```
# getting born clutch id of those in stage 1.1
clutches_of_deads1=dead_wRing_knownDd_blood%>%
  select(BornClutchId)%>%
```

```

distinct(BornClutchId)%>%
na.omit()

#n_distinct(clutches_of_deads1$BornClutchId)

all_sibs=bird%>% # all records of all ids
right_join(clutches_of_deads1)%>% # getting only ids born in the clutches of interest (645 ids)
anti_join(sequenced_ids, by = "RingId")%>% # removing ids already sequenced
anti_join(dead_wRing_knownDd_blood, by = 'RingId') %>% # removing ids from 1.1
anti_join(total_missings, by = 'RingId') # remove ids from stage 1.2

ringed_sibs=
all_sibs%>%
left_join(lab)%>% # combining with lab info
filter(SampleType%in%c("blood", "Blood cell")) # filtering for sibs only with blood samples.

```

Total number with blood samples:

```
n_distinct(ringed_sibs$RingId)
```

```
## [1] 78
```

Number of these with DNA already extracted:

```

DNA_extr_4=big_boy_lab%>%
select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Placement)
rename(RingId=Sample_Name_1)%>%
filter(RingId!='')%>%
inner_join(ringed_sibs, by = 'RingId')%>%
filter(Date.extraction!='' | is.na(Date.extraction))

```

```

## Warning in inner_join(., ringed_sibs, by = "RingId"): Detected an unexpected many-to-many relationship.
## i Row 421 of 'x' matches multiple rows in 'y'.
## i Row 22 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
## "many-to-many" to silence this warning.

```

```
n_distinct(DNA_extr_4$RingId)
```

```
## [1] 77
```

*Benefits* - Completes the family and add power to analysis by being able to discern inbreeding (etc) vs clutch effects. *Costs* - Not directly interested in these individuals

~ 5 ~

**Siblings of the ids that went missing from 2 (n=116)**



```

# getting born clutch id of those in stage 1.1
clutches_of_missings=missings%>%
  select(BornClutchId)%>%
  na.omit()%>%
  distinct(BornClutchId)

#n_distinct(clutches_of_deads1$BornClutchId)

missing_all_sibs=bird%>% # all records of all ids
  right_join(clutches_of_missings)%>% # getting only ids born in the clutches of interest (645 ids)
  anti_join(sequenced_ids, by = "RingId")%>% # removing ids already sequenced
  anti_join(dead_wRing_knownDd_blood, by = 'RingId') %>% # removing ids from 1.1
  anti_join(total_missings, by = 'RingId') # remove ids from stage 1.2

missing_sibs=
  missing_all_sibs%>%
  left_join(lab,relationship = "many-to-many")%>% # combing with lab info
  filter(SampleType%in%c("blood","Blood cell")) # filtering for sibs only with blood samples.

n_distinct(missing_sibs$RingId)

```

```
## [1] 52
```

Number of these with DNA already extracted:

```

DNA_extr_5=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Placement)
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(missing_sibs, by = 'RingId')%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

```

```

## Warning in inner_join(., missing_sibs, by = "RingId"): Detected an unexpected many-to-many relationship
## i Row 23 of 'x' matches multiple rows in 'y'.
## i Row 97 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
##   "many-to-many"' to silence this warning.

```

```
n_distinct(DNA_extr_5$RingId)
```

```
## [1] 52
```

~ 6 ~

## Siblings of previously sequenced individuals that have high inbreeding coefficients

Under the assumption that the ids within a clutch will have similarly high inbreeding coefficients.

```
##Read in inbreeding coefficients
inb=read.table("Inbreeding_depression_owls/All13085_FuniWE_FHBD512g.txt", header = T)%>%
  rename(RingId=INDVs)
#top 90% inbred individuals from 1st round of sequencing
top_ibcs=inb%>%
  filter(FuniWE > quantile(FuniWE, 0.9))%>% # top % ibcs
  left_join(bird)%>% ## joining with bird info to get DOB
  filter(as.Date(HatchDate, format = "%d/%m/%Y") > as.Date("2010-01-01")) # hatched after / in 2010

#n_distinct(top_ibcs$BornClutchId) ## number of different clutches that have highest ibcs

#inbred clutches
inbred_clutches=top_ibcs%>%
  select(BornClutchId)%>%
  distinct(BornClutchId)%>%
  na.omit()
# inbred sibling within those inbred clutches
inbred_sibs=bird%>%
  right_join(inbred_clutches)%>% #
  anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced
  anti_join(dead_wRing_knownDd_blood, by = 'RingId')%>% # remove ids from stage 1.1
  anti_join(total_missings, by = 'RingId')%>% # remove ids from stage 1.2
  anti_join(ringed_sibs, by = 'RingId')%>% # remove ids from stage 1.4
  left_join(lab, by = "RingId")%>% # join with lab info
  filter(SampleType%in%c("blood", "Blood cell")) #filter for only ids with blood samples
```

Total number with blood samples:

```
n_distinct(inbred_sibs$RingId) # number of samples
```

```
## [1] 75
```

Number of these with DNA already extracted:

```
DNA_extr_6=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_Placement)
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(ringed_sibs, by = 'RingId', relationship = "many-to-many")%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_6$RingId)
```

```
## [1] 77
```

~ 7 ~

Genetic parents of individuals already sequenced last round (n=67)

- Born after 2010

```

seqd_clutch_parents=bird%>%
  right_join(sequenced_ids)%>%
  select(RingId, BornClutchId)%>%
  rename(Id=BornClutchId)%>%
  left_join(clutch)%>%
  filter(Season>2010)%>%
  select(FemaleRing, MaleRing)%>%
  pivot_longer(cols=1:2, values_to = 'RingId')%>%
  na.omit()%>%
  filter(RingId!='')

## check those that have already been done and those that have already been proposed

prop_clutch_parents=seqd_clutch_parents%>%
  select(RingId)%>%
  anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced
  anti_join(dead_wRing_knownDd_blood, by = 'RingId')%>% # remove ids from stage 1.1
  anti_join(total_missings, by = 'RingId')%>% # remove ids from stage 1.2
  anti_join(ringed_sibs, by = 'RingId')%>%
  anti_join(inbred_sibs, by = 'RingId')%>%
  unique()%>%
  left_join(lab, by = "RingId")%>% # join with lab info
  filter(SampleType%in%c("blood", "Blood cell"))

```

Total number with blood samples:

```
n_distinct(prop_clutch_parents$RingId)
```

```
## [1] 67
```

Number of these with DNA already extracted:

```

DNA_extr_7=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Placement)
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(prop_clutch_parents, by = 'RingId', relationship = "many-to-many")%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_7$RingId)

```

```
## [1] 65
```

~ 8 ~

Genetic parents of individuals proposed in 1 (n=16) and 2 (n=9)

- Born after 2010
- Many were actually sequenced last time round

## 8.1

```
p1_1_seqd_clutch_parents=dead_wRing_knownDd_blood%>%
  select(RingId, BornClutchId)%>%
  rename(Id=BornClutchId)%>%
  left_join(clutch)%>%
  filter(Season>2010)%>%
  select(FemaleRing, MaleRing)%>%
  pivot_longer(cols=1:2, values_to = 'RingId')%>%
  na.omit()%>%
  filter(RingId!='')

## check those that have already been done and those that have already been proposed

p1_1_prop_clutch_parents=p1_1_seqd_clutch_parents%>%
  select(RingId)%>%
  anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced
  anti_join(dead_wRing_knownDd_blood, by = 'RingId')%>% # remove ids from stage 1.1
  anti_join(total_missings, by = 'RingId')%>% # remove ids from stage 1.2
  anti_join(ringed_sibs, by = 'RingId')%>%
  anti_join(inbred_sibs, by = 'RingId')%>%
  unique()%>%
  left_join(lab, by = "RingId")%>% # join with lab info
  filter(SampleType%in%c("blood", "Blood cell"))

n_distinct(p1_1_prop_clutch_parents$RingId)
```

```
## [1] 16
```

Number of these with DNA already extracted:

```
DNA_extr_8.1=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Pla
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(p1_1_prop_clutch_parents, by = 'RingId', relationship = "many-to-many")%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_8.1$RingId)
```

```
## [1] 16
```

## 8.2

```
p1_2_seqd_clutch_parents=total_missings%>%
  select(RingId, BornClutchId)%>%
  rename(Id=BornClutchId)%>%
  left_join(clutch)%>%
  filter(Season>2010)%>%
```

```
select(FemaleRing, MaleRing)%>%
pivot_longer(cols=1:2, values_to = 'RingId')%>%
na.omit()%>%
filter(RingId!='')
```

```
## Joining with 'by = join_by(Id)'
```

```
## check those that have already been done and those that have already been proposed
```

```
p1_2_prop_clutch_parents=p1_2_seqd_clutch_parents%>%
select(RingId)%>%
anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced
anti_join(dead_wRing_knownDd_blood, by = 'RingId')%>% # remove ids from stage 1.1
anti_join(total_missings, by = 'RingId')%>% # remove ids from stage 1.2
anti_join(ringed_sibs, by = 'RingId')%>%
anti_join(inbred_sibs, by = 'RingId')%>%
unique()%>%
left_join(lab, by = "RingId")%>% # join with lab info
filter(SampleType%in%c("blood", "Blood cell"))
```

```
n_distinct(p1_2_prop_clutch_parents$RingId)
```

```
## [1] 9
```

Number of these with DNA already extracted:

```
DNA_extr_8.2=big_boy_lab%>%
select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Pla
rename(RingId=Sample_Name_1)%>%
filter(RingId!='')%>%
inner_join(p1_2_prop_clutch_parents, by = 'RingId', relationship = "many-to-many")%>%
filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_8.2$RingId)
```

```
## [1] 9
```

~ 9 ~

**Siblings of the ids from 3 (n=52) and their parents (n=10)**

(again a lot were sequenced last time and are also in other lists)

Sibs

```
clutches_of_norings=all_unringed_linked_blood%>%
select(BornClutchId)%>%
na.omit()%>%
distinct(BornClutchId)
```

```

#n_distinct(clutches_of_deads1$BornClutchId)

norings_all_sibs=bird%>% # all records of all ids
  right_join(clutches_of_norings)%>% # getting only ids born in the clutches of interest (645 ids)
  anti_join(sequenced_ids, by = "RingId")%>% # removing ids already sequenced
  anti_join(dead_wRing_knownDd_blood, by = 'RingId') %>%# removing ids from 1.1
  anti_join(total_missings, by = 'RingId') # remove ids from stage 1.2

norings_sibs=
  norings_all_sibs%>%
  left_join(lab,relationship = "many-to-many")%>% # combing with lab info
  filter(SampleType%in%c("blood","Blood cell")) # filtering for sibs only with blood samples.

n_distinct(norings_sibs$RingId)

```

```
## [1] 52
```

Number of these with DNA already extracted:

```

DNA_extr_9.1=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Pla
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(norings_sibs, by = 'RingId',relationship = "many-to-many")%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_9.1$RingId)

```

```
## [1] 52
```

Parents:

```

#####
### parents #####
p1_3_noring_clutch_parents=all_unringed_linked_blood%>%
  select(BornClutchId)%>%
  rename(Id=BornClutchId)%>%
  left_join(clutch)%>%
  filter(Season>2010)%>%
  select(FemaleRing, MaleRing)%>%
  pivot_longer(cols=1:2, values_to = 'RingId')%>%
  na.omit()%>%
  filter(RingId!='')

```

```
## Joining with 'by = join_by(Id)'
```

```

## check those that have already been done and those that have already been proposed

p1_3_prop_clutch_parents=p1_3_noring_clutch_parents%>%
  select(RingId)%>%

```

```

anti_join(sequenced_ids, by = "RingId")%>% # remove ids previously sequenced
anti_join(dead_wRing_knownDd_blood, by = 'RingId')%>% # remove ids from stage 1.1
anti_join(total_missings, by = 'RingId')%>% # remove ids from stage 1.2
anti_join(ringed_sibs, by = 'RingId')%>%
anti_join(inbred_sibs, by = 'RingId')%>%
unique()%>%
left_join(lab, by = "RingId")%>% # join with lab info
filter(SampleType%in%c("blood", "Blood cell"))

n_distinct(p1_3_prop_clutch_parents$RingId)

```

```
## [1] 10
```

Number of these with DNA already extracted:

```

DNA_extr_9.1=big_boy_lab%>%
  select(Sample_Name_1, Date.extraction, DNA.Freezer.Rack.Box.Localisation, DNA_Box_Name, DNA_Tube_.Pla
  rename(RingId=Sample_Name_1)%>%
  filter(RingId!='')%>%
  inner_join(p1_3_prop_clutch_parents, by = 'RingId', relationship = "many-to-many")%>%
  filter(Date.extraction!='' | is.na(Date.extraction))

n_distinct(DNA_extr_9.1$RingId)

```

```
## [1] 10
```

```
## just checking the total number of ids proposed ##
```

```

missing=total_missings$RingId
dead_ids=dead_wRing_knownDd_blood$RingId
inbred_ids=inbred_sibs$RingId

```

```

all_Ringids_plate_1=c(missing, dead_ids, inbred_ids)%>%
  unique()%>%
  as.data.frame()%>%
  rename(RingId='.')%>%
  mutate(proposed_seq_round_2 = 'yes')

```

```

ringed_sibs_ids=ringed_sibs$RingId
missing_sibs_ids=missing_sibs$RingId
prop_clutch_parents_ids=prop_clutch_parents$RingId
p1_2_prop_clutch_parents_ids=p1_2_prop_clutch_parents$RingId
p1_1_prop_clutch_parents_ids=p1_1_prop_clutch_parents$RingId

```

```

all_Ringids_plate_2=c(ringed_sibs_ids, missing_sibs_ids, prop_clutch_parents_ids, p1_2_prop_clutch_pare
  unique()%>%
  as.data.frame()%>%
  rename(RingId='.')%>%
  anti_join(all_Ringids_plate_1, by = 'RingId')%>%
  mutate(proposed_seq_round_3 = 'yes')

```

```

unringed_proposed=all_unringed_linked_blood%>%
  select(BirdId, HatchDate)%>%
  unique()%>%
  as.data.frame()%>%
  mutate(proposed_seq_round_2_unringed = 'yes')

```

Distribution of proposed individuals, those already sequenced and those left over per year

```

all_ringed_ids_after2010=bird%>%
  mutate(ring_date = as.Date(RingDate, format = "%d/%m/%Y"))%>%
  separate_wider_delim(ring_date, delim = "-", cols_remove=F, names = c("year", NA, NA))%>%
  select(RingId, year)%>%
  unique()%>%
  na.omit()%>%
  filter(year>2009)%>%
  left_join(sequenced_ids)%>%
  left_join(all_Ringids_plate_1)%>%
  left_join(all_Ringids_plate_2)

per_yr=all_ringed_ids_after2010%>%
  group_by(year)%>%
  summarise(total_ringed=n(),
            Sequenced=sum((seq_round_1 == 'yes'), na.rm = TRUE),
            Newly_proposed=sum((proposed_seq_round_2 == 'yes'|proposed_seq_round_3 == 'yes'), na.rm = TRUE),
            mutate(Remainder_indivs=total_ringed-(Sequenced+Newly_proposed))

all_UNringed_ids_after2010=bird%>%
  mutate(hatch_date = as.Date(HatchDate, format = "%d/%m/%Y"))%>%
  separate_wider_delim(hatch_date, delim = "-", cols_remove=F, names = c("year", NA, NA))%>%
  filter(RingId=='')%>%
  select(BirdId, year)%>%
  unique()%>%
  na.omit()%>%
  filter(year>2009)%>%
  left_join(unringed_proposed, by='BirdId')

per_yr_unringed=all_UNringed_ids_after2010%>%
  group_by(year)%>%
  summarise(total_UNringed=n(),
            Newly_proposed_unringed=sum((proposed_seq_round_2_unringed == 'yes'), na.rm = TRUE))%>%
  mutate(Remainder_unringed_indivs=total_UNringed-Newly_proposed_unringed)

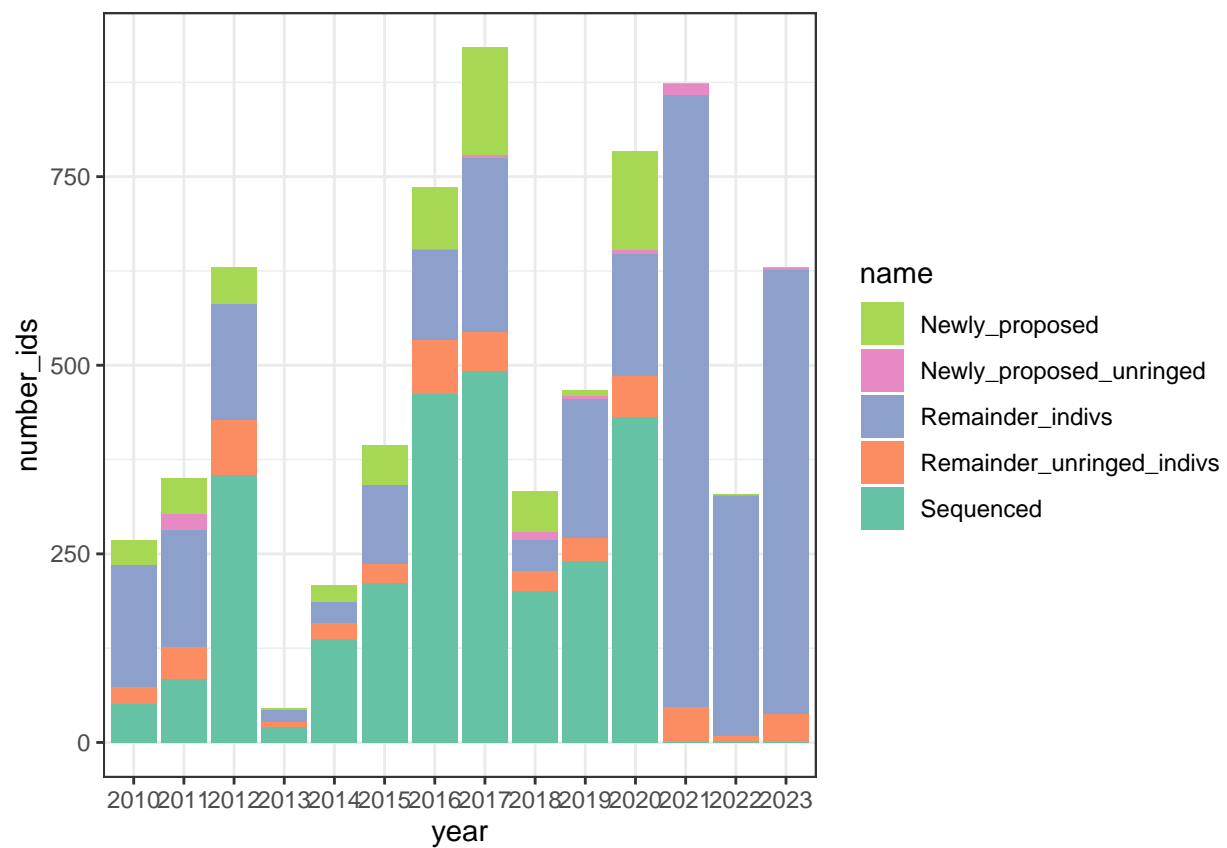
all_birds_after2010=
  merge(per_yr, per_yr_unringed)

```



```
plotting=all_birds_after2010%>%
  pivot_longer(cols=c(Remainder_unringed_indivs, Newly_proposed_unringed, Sequenced, Remainder_indivs, I

plotting%>%
  ggplot(aes(x=year,y=number_ids,fill=name)) +
  geom_bar(position = "stack", stat="identity")+
  theme_bw()+
  scale_fill_brewer(palette = 'Set2', direction = -1)
```



I see now that some of the newly proposed unringed individuals are in the later years which i may remove  
...