Mitochondial mutational spectrum in recombinant inbred mouse lines

Valeriia Timonina

3/18/2022

Background

Based on data from: A natural mutator allele shapes mutation spectrum variation in mice Thomas A. Sasani, David G. Ashbrook, Annabel C. Beichman, Lu Lu, Abraham A. Palmer, Robert W. Williams, Jonathan K. Pritchard, Kelley Harris bioRxiv 2021.03.12.435196; doi: https://doi.org/10.1101/2021.03.12.435196 Two inbred mice strains C57**B**L/6J (B) and **D**BA/2J (D) were crossed to get a pannel of recombinant inbred mouse lines (BXD). Each line was maintatined by brother-sister mating and accommulating *de novo* mutations for up to 50 years. It was shown that mice with D haplotype at a QTL on chr4 accumulate more C>A mutations.B and D haplotypes encode different alleles of the DNA repair gene *Mutyh*. Here we want to check if lines that have D haplotype in *Mutyh* gene also have different mutational spectrum in mitoDNA. Also we want to repeat QTL analysis and see if any other locus influences mtDNA mutational spectrum.

Construction of the BXD RILs

BXD RILs were derived from crosses of C57**B**L/6J and **D**BA/2J inbred laboratory strains initiated in 6 distinct epochs from 1971 to 2014. Two strategies: 1) 4 epoches were produsedusing standart F2 cross where a male DBA/2J mouse is crossed to a **C57BL/6J female** to produce F1 animals that are heterozygous for parental ancestry at essentially all loci in the genome. Pairs of these F1 animals are then crossed to produce F2s. To generate each individual recombinant inbred line, a brother and sister are picked from among the F2s and mated; this brother-sister mating strategy continues for many generations. 2) Two epoches were produced using the advanced intercross strategy. F2s are generated as in the standard F2 cross. However, pseudo-random pairs of F2 animals are then crossed to generate F3s, pseudo-random pairs of F3s are crossed to generate F4s, and so on, for up to 14 generations. Then, to generate inbred lines, brother-sister matings are once again initiated from the offspring of the final pseudo-random cross.

1. Read genotype data

```
Genotypes = read.table('../data/MiceGenotypes/BXD.geno', header=TRUE)

kable(head(Genotypes[,1:10]))
```

Chr	Locus	cM	Mb	BXD1	BXD2	BXD5	BXD6	BXD8	BXD9
1	rs31443144	1.50	3.010274	B	B	D	D	D	B
1	rs6269442	1.50	3.492195	B	B	D	D	D	B
1	rs32285189	1.63	3.511204	B	B	D	D	D	B
1	rs258367496	1.63	3.659804	B	B	D	D	D	B
1	rs32430919	1.75	3.777023	B	B	D	D	D	B
1	rs36251697	1.88	3.812265	B	B	D	D	D	B

The locus in QTL on chr4 with highest LOD is rs52263933.

```
TheMutyhGenotype = Genotypes[Genotypes$Locus == 'rs52263933',]
TheMutyhGenotype <- TheMutyhGenotype %>%
```

Var1	Freq
B	84
D	114
H	37
U	1

Var1	Freq
B	84
D	114

```
  pivot_longer(colnames(TheMutyhGenotype[-(1:4)]), names_to = 'line', values_to = 'genotype')
kable(table(TheMutyhGenotype$genotype)) %>%
  kable_styling(full_width = F)
```

Remove H and U for now.

```
TheMutyhGenotype <- TheMutyhGenotype[TheMutyhGenotype$genotype %in% c('B', 'D'),]
kable(table(TheMutyhGenotype$genotype))%>%
  kable_styling(full_width = F)
```

2. Read mtDNA variants and add TheMutyhGenotypeDataFrame column

---

```
MtDna = read.table('../data/all_annmax_single.csv', sep = ';', header=TRUE)
```

```
MtDna$SNP = gsub(">",'',MtDna$SNP)
MtDna$NameNew = gsub("\\_(.*)",'',MtDna$Name) #  BXD71_RwwJ > BXD71
MtDna$NameNew = gsub("BXD0","BXD",MtDna$NameNew) #  BXD009 > BXD09
MtDna$NameNew = gsub("BXD0","BXD",MtDna$NameNew) #  BXD09 > BXD9
MtDna$NameNew = gsub("BXD0","BXD",MtDna$NameNew) #  BXD09 > BXD9

length(intersect(sort(unique(MtDna$NameNew)), TheMutyhGenotype$line))
```

```
## [1] 89
```

89 common lines out of 100 in mt DNA variants data and 198 in genotypes data. Which do not intersect?

```
length(setdiff(TheMutyhGenotype$line, unique(MtDna$NameNew)))
```

```
## [1] 109
```

```
setdiff(unique(MtDna$NameNew), TheMutyhGenotype$line)
```

```
##  [1] "C57BL"  "DBA"    "BXD199" "BXD227" "BXD131" "BXD221" "BXD224" "BXD127"
##  [9] "BXD222" "BXD217" "BXD215"
```

```
length(setdiff(unique(MtDna$NameNew), TheMutyhGenotype$line))
```

```
## [1] 11
```

3. Merge data

```
MtDna = merge(MtDna,TheMutyhGenotype, by.x = 'NameNew', by.y = 'line', all.x = TRUE)
kable(table(MtDna$genotype))%>%
  kable_styling(full_width = F)
```

Var1	Freq
B	3524
D	5554

4. Filtration of data to get normal expected mutspec

Each variant can be:

1) difference of ancestal line from RefSeq (should be very common);
2) inherited heteroplasmy (should be more similar in close inbred lines ~ recently diverged lines from the same epoch);
3) de novo (recent) mutation
4) noise (sequence artefacts)

Filtration steps:

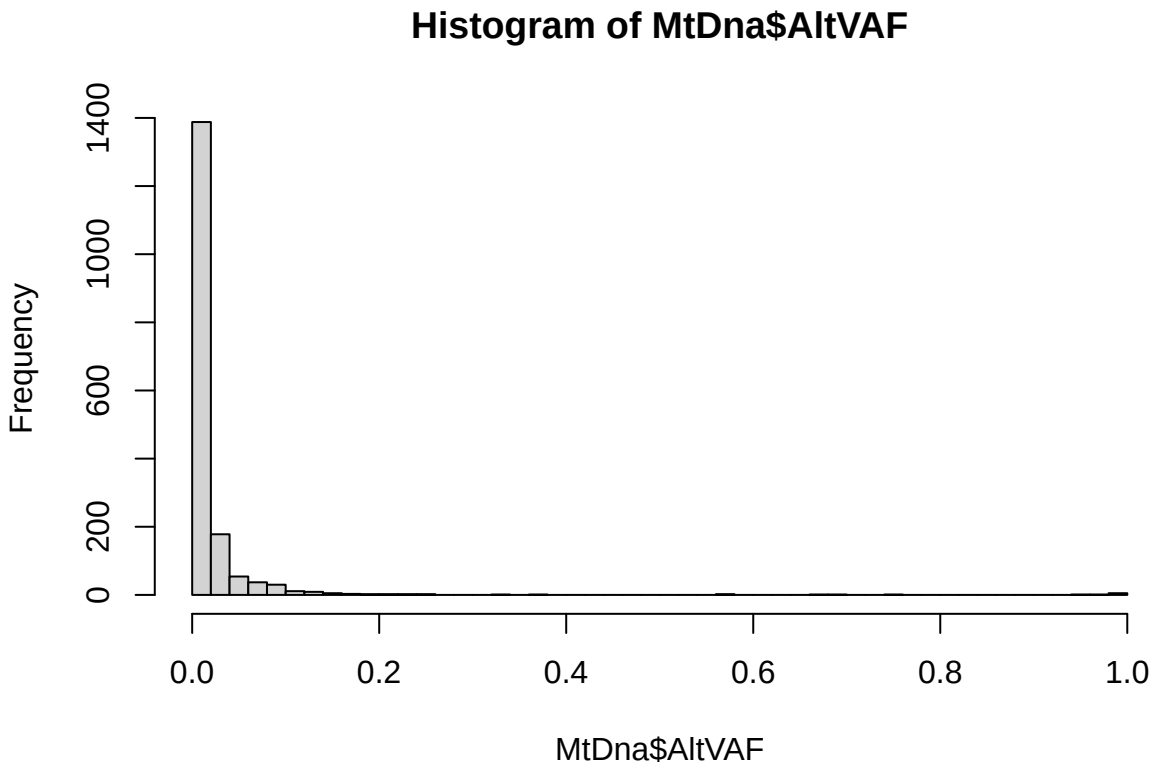
- The sum of reads with mutation should be >= 10
- The number of each forvard and reverse reads > 4
- The same for reference reads

```
MtDna <- separate(data = MtDna, col = Allele_specific_forward_reverse_read_counts, sep = '\\||,', into =

MtDna[,c('RefF', 'RefR', 'AltF', 'AltR')] <- apply(MtDna[,c('RefF', 'RefR', 'AltF', 'AltR')], 2, as.nume

MtDna = MtDna[(MtDna$AltF + MtDna$AltR) >= 10 & MtDna$AltR > 4 & MtDna$AltF > 4 & (MtDna$RefF + MtDna$Re
```

1737 mutations left

Let's have a look at the distribution of VAF mutations

```
MtDna <- separate(MtDna, col = variant_allele_frequency, sep = '\\|', into = c('RefVAF', 'AltVAF'))
MtDna[,c('RefVAF', 'AltVAF')] <- apply(MtDna[,c('RefVAF', 'AltVAF')], 2, as.numeric)

hist(MtDna$AltVAF, breaks = 50)
```

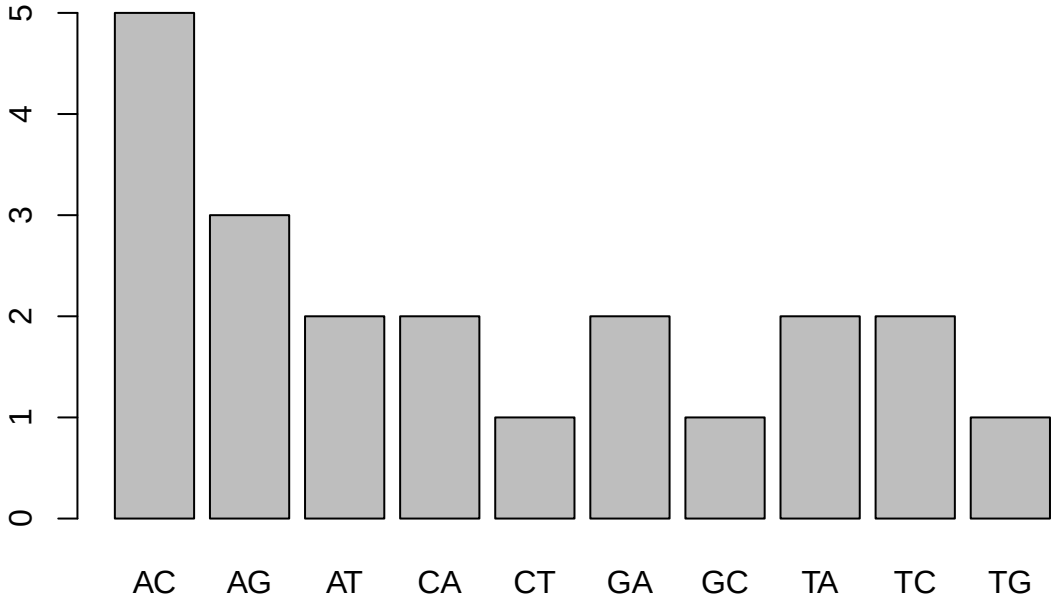



```
summary(MtDna$AltVAF)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001863 0.005372 0.009171 0.023382 0.016103 0.995787
```

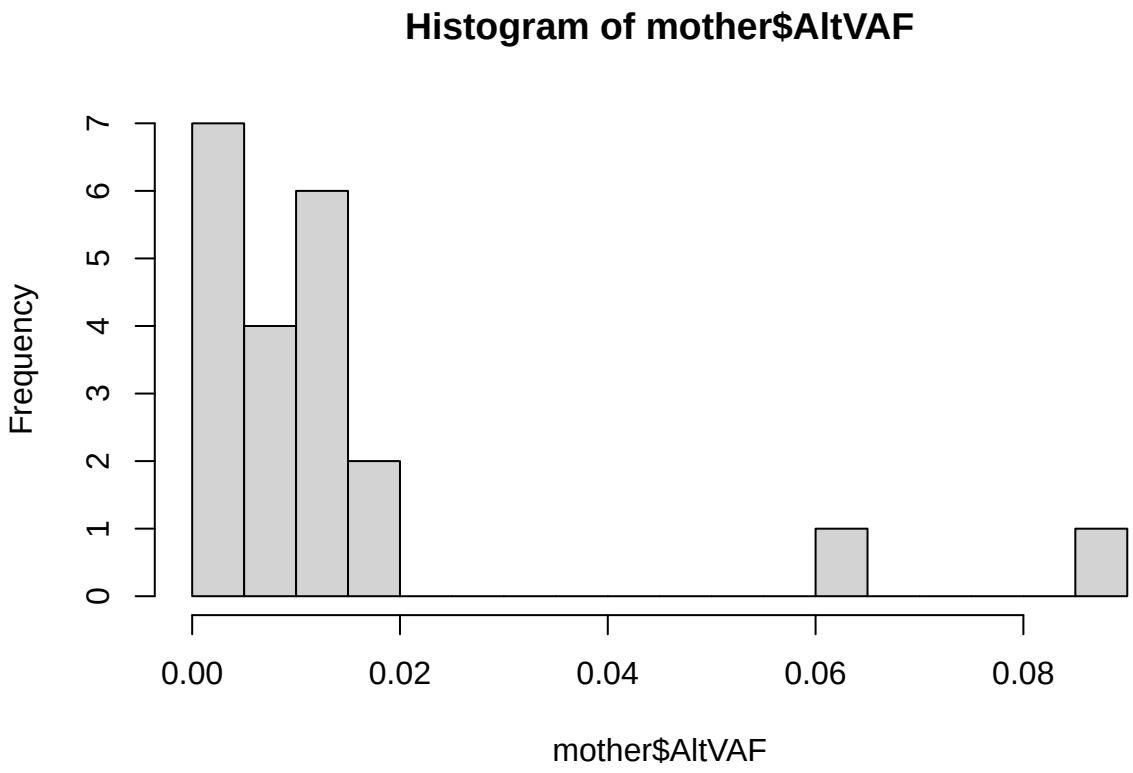
The mitochondrial mouse Eva is C57BL/6J strain so all mitochondria in RILs are its progeny.

```
mother <- MtDna[MtDna$NameNew == 'C57BL',]
barplot(table(mother$SNP))
```



SHOULD WE DELETE ALL MUTATIONS THAT ARE IN C57BL?

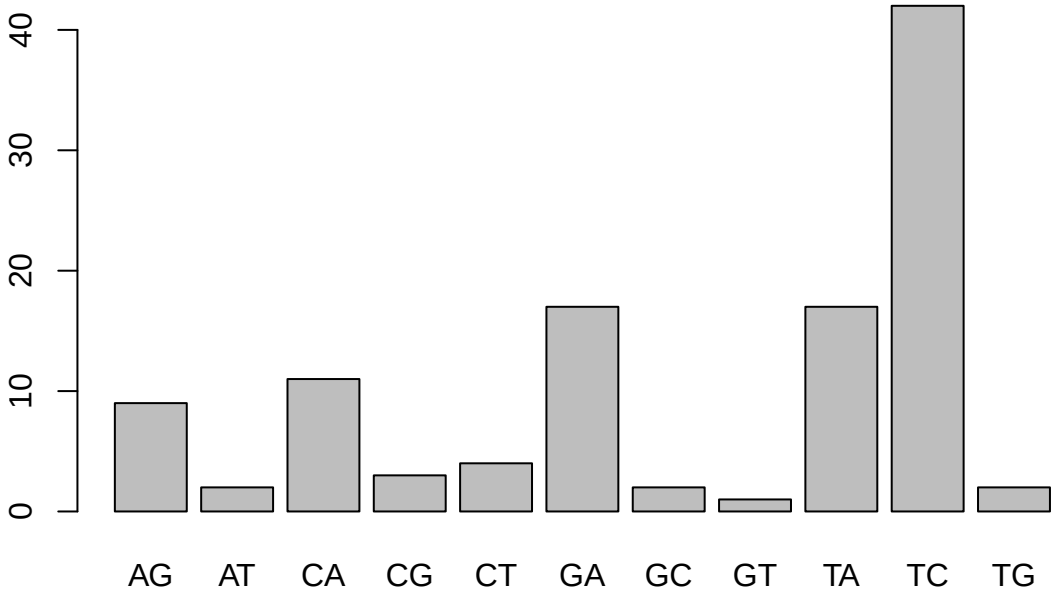
```
hist(mother$AltVAF, breaks = 20)
```




```
MtDna$MutID <- paste(MtDna$Pos, MtDna$SNP, sep = '_')
length(unique(MtDna$MutID))
```

```
## [1] 646
```

```
MtDnaShort = MtDna[!(MtDna$MutID %in% MtDna[MtDna$NameNew == 'C57BL',]$MutID),] # 21 > 1737-1420
MtDnaShort2 = MtDna[!(MtDna$MutID %in% MtDna[(MtDna$NameNew == 'C57BL') & (MtDna$AltVAF > 0.05),]$MutID)

barplot(table(MtDnaShort[MtDnaShort$AltVAF >= 0.03,]$SNP))
```



Singletones

```
mutID_Freq <- data.frame(table(MtDnaShort$MutID))
hist(mutID_Freq$Freq, breaks = 50)
```