Donor Data Stats

Kate Weaver

10/16/2021

Load packages and dataframe

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(here)
```

```
## here() starts at /Users/kateweaver/mccoyLab_withOthers/transmission-distortion
```

```
library(patchwork)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
load("bell_true_com_res_withInput.Rdata")
colnames(full_df)
```

```
## [1] "sample"      "sample_desc" "chr"         "num_snps"    "num_gametes"
## [6] "phase_com"   "imp_com"     "input_com"   "res"
```

```
inf_df <- full_df[full_df$sample_desc == "inf",]
ninf_df <- full_df[full_df$sample_desc == "ninf",]
```

Look at number of gametes

```
donor_spec_gam <- lapply(1:length(unique(full_df$sample)), function(x) full_df$num_gametes[which(unique
donor_spec_gam
```

```
## $nc1abnov17
##  [1] 981 980 982 982 982 981 982 981 980 982 982 982 980 980 981 982 981 982 982
## [20] 982 979 981
##
## $nc2absept17
##  [1] 1677 1674 1674 1678 1678 1677 1679 1677 1666 1680 1680 1679 1679 1679 1678
## [16] 1678 1680 1679 1679 1679 1679 1679
##
## $nc3aboct17
##  [1] 1288 1287 1287 1286 1288 1289 1288 1288 1284 1289 1288 1288 1288 1288 1288
## [16] 1288 1288 1289 1287 1288 1288 1282
##
## $nc4abnov17
##  [1] 1479 1482 1482 1480 1480 1481 1481 1482 1481 1482 1482 1482 1482 1482 1482
## [16] 1479 1482 1481 1482 1482 1479 1479
##
## $nc6abcd
##  [1] 1365 1369 1370 1369 1370 1370 1370 1370 1368 1370 1370 1370 1369 1369 1370
## [16] 1370 1370 1370 1370 1370 1368 1366
##
## $nc8ab
##  [1] 1659 1663 1659 1659 1663 1662 1661 1663 1660 1660 1662 1663 1654 1659 1658
## [16] 1661 1663 1660 1662 1662 1656 1660
##
## $nc9ab
##  [1] 1888 1892 1894 1893 1892 1893 1892 1893 1892 1893 1894 1894 1893 1894 1892
## [16] 1894 1894 1894 1894 1894 1891 1892
##
## $nc10oldoil
##  [1] 1154 1151 1154 1153 1154 1153 1154 1154 1149 1153 1154 1152 1154 1153 1153
## [16] 1154 1154 1153 1153 1153 1149 1153
##
## $nc11ab
##  [1] 1927 1928 1928 1928 1930 1930 1928 1929 1927 1930 1930 1930 1926 1929 1925
## [16] 1928 1928 1929 1929 1928 1926 1928
##
## $nc12ab
##  [1] 2144 2143 2144 2143 2144 2145 2143 2144 2141 2145 2145 2144 2144 2142 2142
## [16] 2145 2144 2144 2145 2144 2141 2144
##
## $nc13ab
##  [1] 1512 1513 1514 1513 1513 1514 1514 1514 1511 1514 1513 1513 1514 1513 1513
## [16] 1513 1514 1514 1514 1514 1513 1512
##
## $nc14ab
##  [1] 1336 1335 1336 1333 1335 1335 1335 1334 1334 1335 1335 1335 1333 1336 1332
## [16] 1335 1335 1335 1336 1335 1330 1330
##
## $nc15ab
##  [1] 1701 1700 1701 1702 1702 1702 1700 1700 1701 1702 1701 1702 1700 1698 1699
## [16] 1701 1702 1701 1701 1700 1700 1702
```

```
##
## $nc16ab
##  [1] 1783 1782 1784 1781 1785 1785 1784 1784 1780 1781 1784 1784 1780 1782 1784
## [16] 1780 1783 1783 1784 1783 1783 1784
##
## $nc17ab
##  [1] 1502 1501 1503 1503 1502 1504 1503 1504 1502 1504 1503 1504 1503 1503 1502
## [16] 1502 1504 1501 1502 1503 1499 1500
##
## $nc18ab
##  [1] 1589 1588 1588 1588 1587 1589 1589 1589 1586 1588 1588 1588 1587 1589 1588
## [16] 1589 1588 1588 1588 1589 1586 1587
##
## $nc22abcd
##  [1] 1692 1692 1692 1692 1692 1692 1691 1691 1689 1692 1693 1693 1693 1693 1691
## [16] 1692 1693 1693 1693 1691 1690 1687
##
## $nc25abcd
##  [1] 2272 2270 2274 2273 2272 2272 2272 2273 2258 2274 2272 2274 2270 2272 2262
## [16] 2271 2272 2273 2273 2274 2266 2268
##
## $nc26abcd
##  [1] 974 973 974 974 974 974 974 974 974 973 974 974 974 974 973 972 974 974 974
## [20] 974 971 969
##
## $nc27aboct17
##  [1] 1264 1262 1265 1268 1268 1268 1268 1267 1267 1268 1268 1268 1268 1264 1265
## [16] 1267 1267 1268 1264 1267 1268 1268
##
## $ff3a
##  [1] 3929 3929 3929 3929 3929 3929 3929 3928 3929 3929 3929 3929 3929 3926 3927
## [16] 3929 3929 3929 3929 3928 3926 3929
##
## $ff4a
##  [1] 1410 1410 1410 1410 1410 1410 1410 1410 1410 1410 1410 1410 1410 1410 1410
## [16] 1410 1410 1410 1410 1410 1410 1410
##
## $pb2a
##  [1] 1523 1523 1521 1523 1523 1523 1523 1523 1523 1523 1523 1523 1523 1523 1523
## [16] 1523 1523 1523 1523 1523 1523 1523
##
## $pb3a
##  [1] 2289 2289 2289 2289 2289 2289 2289 2289 2289 2289 2289 2289 2289 2289 2289
## [16] 2289 2289 2289 2288 2289 2289 2288
##
## $pb4a
##  [1] 2421 2421 2421 2421 2421 2421 2421 2421 2421 2421 2421 2421 2421 2421 2421
## [16] 2421 2421 2421 2421 2421 2420 2420
```

```
unlist(lapply(1:length(donor_spec_gam), function(x) length(unique(donor_spec_gam[[x]]))  == 1)) %>% `na
```

```
##  nc1abnov17 nc2absept17  nc3aboct17  nc4abnov17     nc6abcd       nc8ab
##       FALSE       FALSE       FALSE       FALSE       FALSE       FALSE
##       nc9ab  nc10oldoil      nc11ab      nc12ab      nc13ab      nc14ab
##       FALSE       FALSE       FALSE       FALSE       FALSE       FALSE
```

```
##      nc15ab      nc16ab      nc17ab      nc18ab    nc22abcd    nc25abcd
##       FALSE       FALSE       FALSE       FALSE       FALSE       FALSE
##    nc26abcd nc27aboct17        ff3a        ff4a        pb2a        pb3a
##       FALSE       FALSE       FALSE        TRUE       FALSE       FALSE
##        pb4a
##       FALSE
```

Note that all donors but ff4a have differing numbers of gametes per chromosome. Could this be due to filtering to just euploid data?

```
num_gams <- data.frame()
for (i in 1:22){
  num_gams <- rbind(num_gams, data.frame(chr=paste0("chr", i), num_cells=sum(unlist(lapply(1:length(nam
}
num_gams
```

```
##      chr num_cells
## 1   chr1     42759
## 2   chr2     42757
## 3   chr3     42775
## 4   chr4     42770
## 5   chr5     42783
## 6   chr6     42788
## 7   chr7     42780
## 8   chr8     42782
## 9   chr9     42722
## 10 chr10     42787
## 11 chr11     42790
## 12 chr12     42791
## 13 chr13     42763
## 14 chr14     42768
## 15 chr15     42748
## 16 chr16     42773
## 17 chr17     42788
## 18 chr18     42783
## 19 chr19     42783
## 20 chr20     42783
## 21 chr21     42730
## 22 chr22     42741
```

Given the observation before this that gamete number isn't constant across chromosomes for a given donor, here I've summed the number of gametes across the donors in a chromosome specific manner. chr9 has the minimum number of 42722 cells and chr12 has the maximum with 42791 cells. Note this is with pooling the infertile donors with the original set.

```
mean(num_gams$num_cells)
```

```
## [1] 42770.18
```

Infertile Donors

```
min_inf_gam <- which(inf_df$num_gametes == min(inf_df$num_gametes))[1]
inf_df[min_inf_gam, c("sample", "chr", "num_gametes")]
```

```
##    sample chr num_gametes
## 22   ff4a   1        1410
```

```
max_inf_gam <- which(inf_df$num_gametes == max(inf_df$num_gametes))[1]
inf_df[max_inf_gam, c("sample", 'chr', "num_gametes")]
```

```
##    sample chr num_gametes
## 21   ff3a   1        3929
```

```
quant_inf <- quantile(inf_df$num_gametes)
quant_inf
```

```
##   0%  25%  50%  75% 100%
## 1410 1523 2289 2421 3929
```

Original Cohort of Donors

```
min_ninf_gam <- which(ninf_df$num_gametes == min(ninf_df$num_gametes))[1]
ninf_df[min_ninf_gam, c("sample",'chr', "num_gametes")]
```

```
##       sample chr num_gametes
## 544 nc26abcd  22         969
```

```
max_ninf_gam <- which(ninf_df$num_gametes == max(ninf_df$num_gametes))[1]
ninf_df[max_ninf_gam, c("sample",'chr', "num_gametes")]
```

```
##      sample chr num_gametes
## 68 nc25abcd   3        2274
```

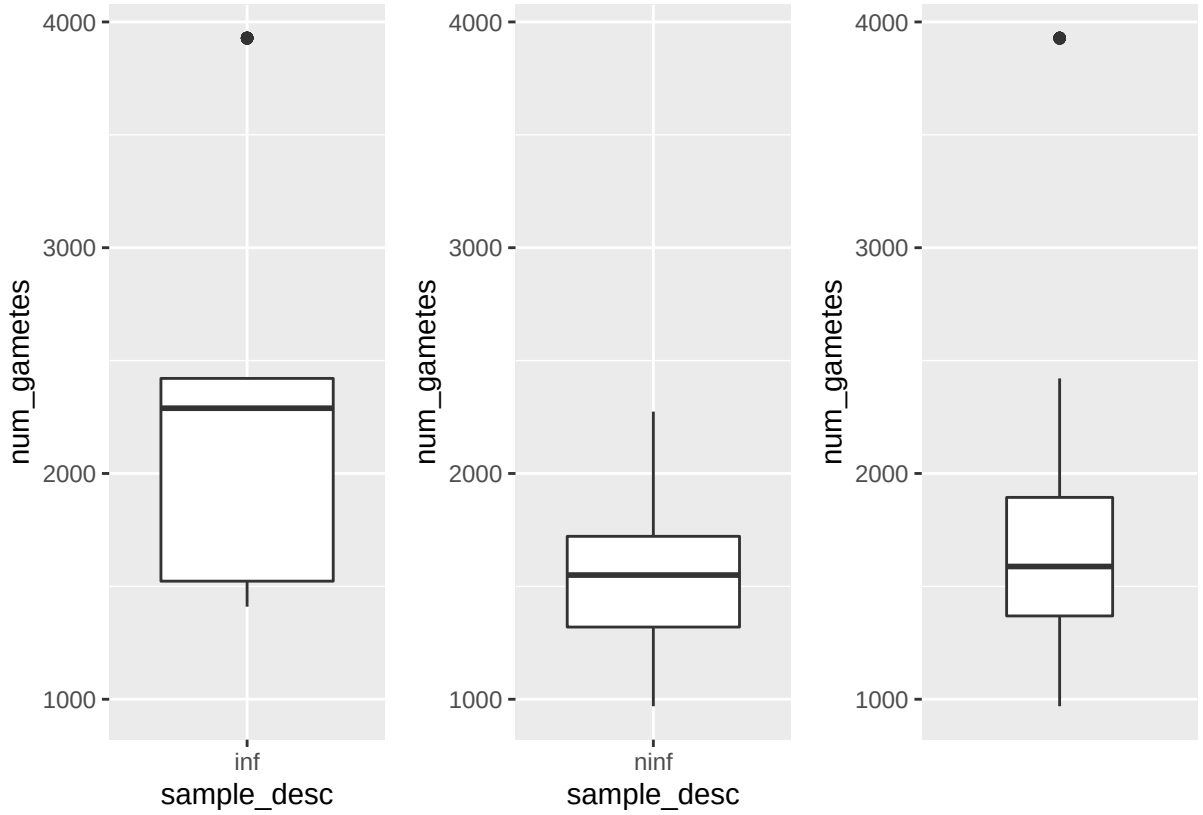
```
quant_ninf <- quantile(ninf_df$num_gametes)
quant_ninf
```

```
##      0%     25%     50%     75%    100%
##  969.00 1319.75 1550.00 1721.50 2274.00
```

```
quant_tot <- quantile(full_df$num_gametes)
quant_tot
```

```
##   0%  25%  50%  75% 100%
##  969 1369 1588 1894 3929
```

```
g_inf <- ggplot(inf_df, aes(x = sample_desc, y = num_gametes)) + geom_boxplot() + scale_y_continuous(li
g_ninf <- ggplot(ninf_df, aes(x = sample_desc, y = num_gametes)) + geom_boxplot() + scale_y_continuous(
g_tot <- ggplot(full_df, aes(y = num_gametes)) + geom_boxplot() + scale_y_continuous(limits=c(min(ninf_

g <- g_inf + g_ninf + g_tot
g
```



In this plot of the number of gametes (donor and chromosome specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left.

```
message(paste0("mean infertile donors: ", mean(inf_df$num_gametes)))
```

```
## mean infertile donors: 2314.25454545455
```

```
message(paste0("sd infertile donors: ", sd(inf_df$num_gametes)))
```

```
## sd infertile donors: 905.44161698485
```

```
message(paste0("mean original donors: ", mean(ninf_df$num_gametes)))
```

```
## mean original donors: 1559.94545454545
```

```
message(paste0("sd original donors: ", sd(ninf_df$num_gametes)))
```

```
## sd original donors: 340.088427101077
```

```
message(paste0('mean all: ', mean(full_df$num_gametes)))
```

```
## mean all: 1710.80727272727
```

```
message(paste0("sd all: ", sd(full_df$num_gametes)))
```

```
## sd all: 588.60785955776
```

Look at number of SNPs post filtering

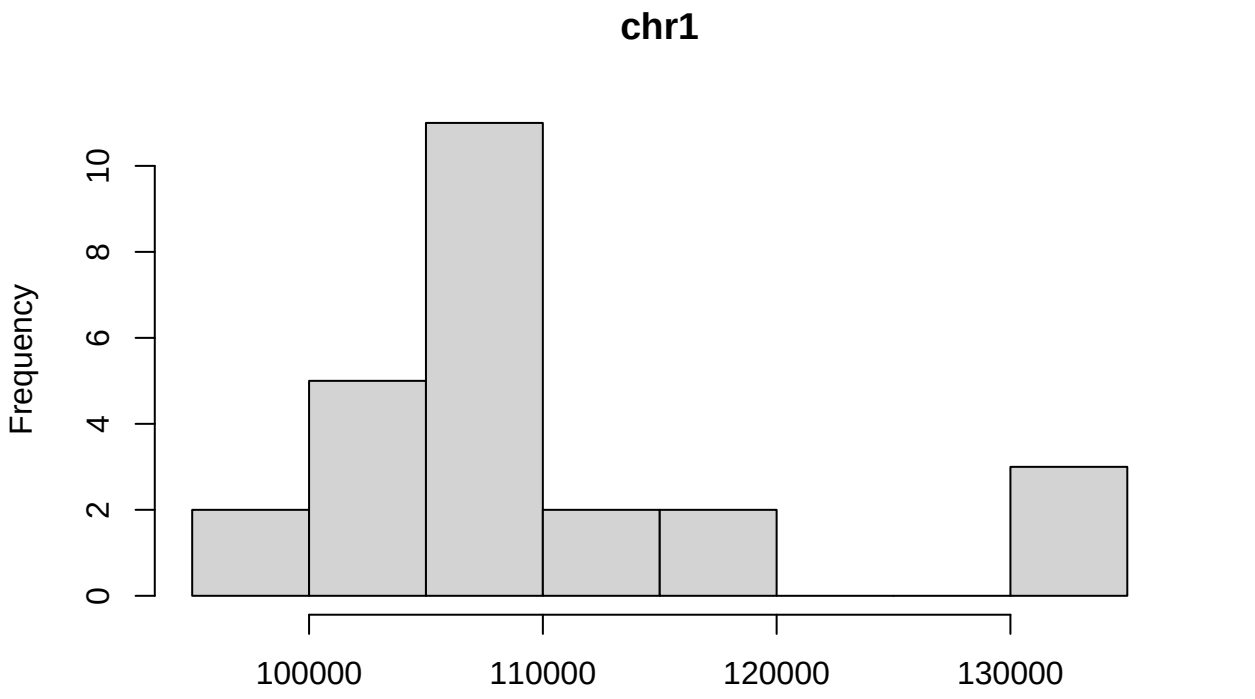
```
donor_spec_snp <- lapply(1:length(unique(full_df$sample)), function(x) full_df$num_snps[which(unique(ful
donor_spec_snp
```

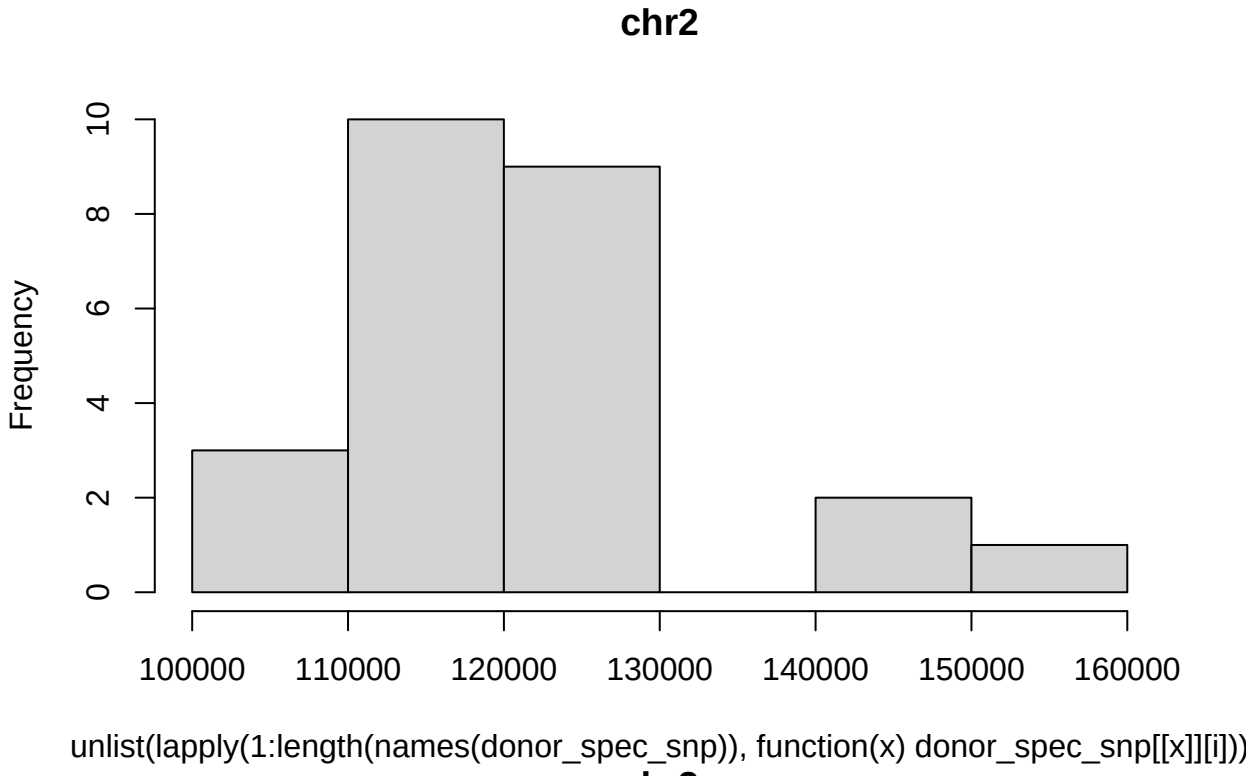
```
## $nc1abnov17
##  [1] 106435 123116 101925 101841  95017  85243  78799  80365  59270  70367
## [11]  67961  65507  49790  44489  38317  42833  33966  41510  26926  33390
## [21]  19314  16509
##
## $nc2absept17
##  [1] 111548 117802  99788 102536  91721  86757  76724  77109  60404  72386
## [11]  67644  64923  52595  43583  40373  43074  33172  38967  28024  33361
## [21]  19205  18821
##
## $nc3aboct17
##  [1] 109095 117054 103411  99468  85925  85258  79759  74330  59676  70568
## [11]  64069  61488  47412  44806  37407  44391  32602  39486  26266  32445
## [21]  18660  17733
##
## $nc4abnov17
##  [1] 106123 120404 105998 102968  88347  87473  79230  73698  57384  69624
## [11]  66241  70500  48977  47315  39837  43989  31503  39427  27079  33594
## [21]  19361  16469
##
## $nc6abcd
##  [1] 131131 142362 121216 118307 110024 102287  92831  93614  72108  84484
## [11]  79479  78011  60541  54011  48541  53821  42235  46259  33027  39491
## [21]  23177  21316
##
## $nc8ab
##  [1]  99843 111278  95434  91643  77260  80562  74014  73040  58623  69837
## [11]  65814  61634  46991  43069  35894  38977  30764  36133  24863  31408
## [21]  18510  14450
##
## $nc9ab
##  [1] 103370 113219  93947  94713  80819  79492  71744  74473  57199  69855
## [11]  63031  62466  48797  42454  35257  39452  30571  38904  24043  29041
## [21]  17176  15182
##
## $nc10oldoil
##  [1] 100948 106351  91243  93964  81269  81010  74245  73394  52409  62700
## [11]  61891  60206  45182  39621  33718  41034  29400  37322  24395  29663
## [21]  16561  15654
##
## $nc11ab
##  [1] 106365 116217 104721 100819  92474  87108  79589  78110  61807  67264
## [11]  70212  65450  50449  45101  37238  42208  34596  40218  28327  31720
## [21]  18217  16255
##
## $nc12ab
##  [1] 102634 114728 101165  98575  90861  85885  78081  70756  59867  69301
## [11]  66961  58827  51876  44762  40130  41057  30576  40481  27048  32069
## [21]  17664  18074
##
## $nc13ab
##  [1] 109025 122544 102846 103408  91465  87405  78561  77266  60102  72160
## [11]  65040  62808  50384  41450  36619  41562  33724  41127  25975  31407
## [21]  18797  16251
```

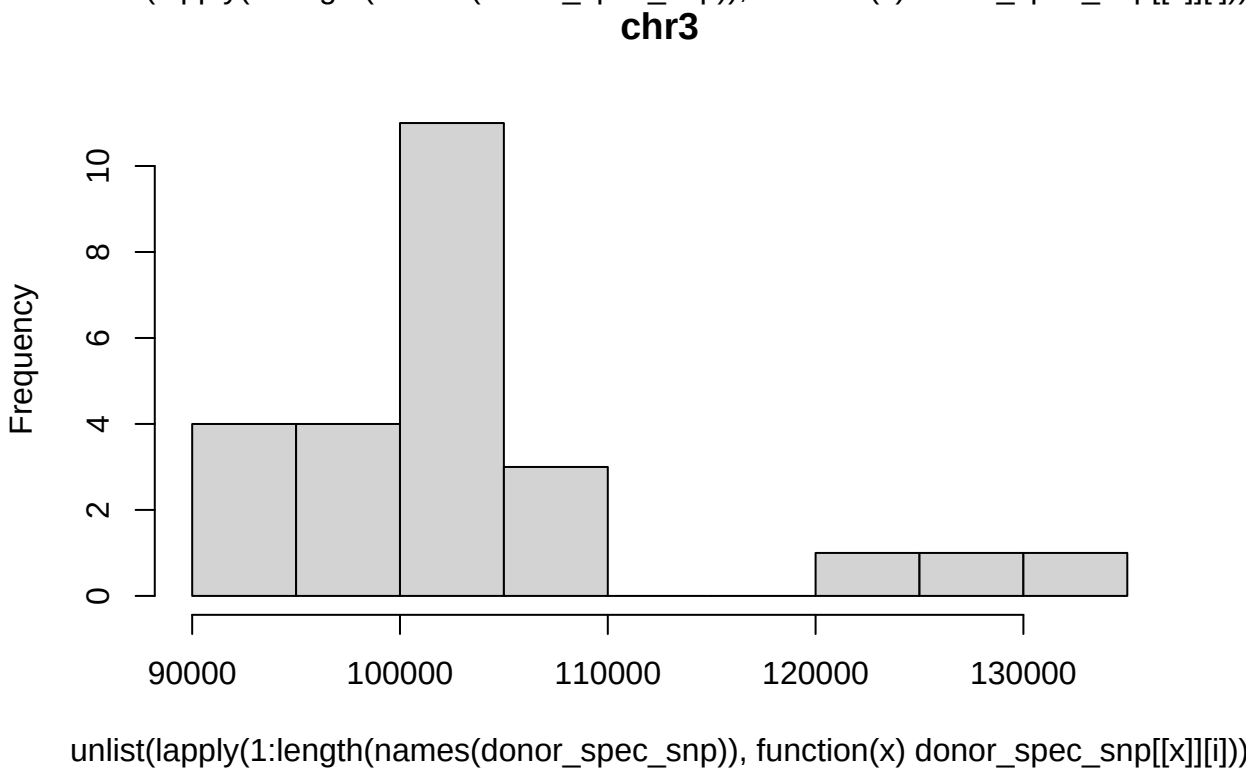
```
##
## $nc14ab
##  [1] 104240 121652 103450 103251  92847  85223  79210  79491  61968  73279
## [11]  64781  66669  50322  44344  38350  42261  34276  41141  26949  30333
## [21]  19272  16149
##
## $nc15ab
##  [1] 107545 120671 102064 104092  86801  87475  76877  78552  58682  72679
## [11]  68938  63433  51748  43885  38234  43948  33333  38764  27135  31358
## [21]  18612  16155
##
## $nc16ab
##  [1] 109308 118831 103106 102961  89712  86836  84465  78349  60591  73329
## [11]  66827  65607  50014  46541  40277  41423  31200  39389  26415  30124
## [21]  19336  16969
##
## $nc17ab
##  [1] 109110 117849  99325 100015  86930  90834  71266  75420  59965  70028
## [11]  66398  64319  51154  46097  38006  43237  30864  38457  25990  33002
## [21]  19914  16833
##
## $nc18ab
##  [1] 108172 114721 102377 100203  87533  84915  80827  69507  58161  72148
## [11]  68353  64383  49368  43872  35681  45021  32046  39147  25413  32202
## [21]  20065  17600
##
## $nc22abcd
##  [1] 134438 144899 126695 120882 109677 105849  98124  97563  74991  87659
## [11]  84974  76487  62634  56331  47253  53828  42537  47570  32670  40976
## [21]  24209  21303
##
## $nc25abcd
##  [1] 130097 156043 131927 132466 108054 109950  98841  99266  78914  94581
## [11]  83626  83215  63280  58793  52250  55681  43801  52626  34647  41638
## [21]  23882  19399
##
## $nc26abcd
##  [1] 117736 123624  92135 105210  90715  91618  86302  79569  63821  75322
## [11]  76543  67951  49769  48278  40060  48268  39498  42208  31228  36474
## [21]  18156  19634
##
## $nc27aboct17
##  [1] 115961 123955 104857 102924  89042  77256  78699  80322  59885  72490
## [11]  70157  67755  51942  46383  40914  43717  33493  41294  27073  30865
## [21]  18480  16696
##
## $ff3a
##  [1]  98659 106197  93906  88459  82843  81125  72956  70027  57269  65718
## [11]  59249  57211  46317  40814  37711  39822  30984  37502  24975  29166
## [21]  18149  14103
##
## $ff4a
##  [1] 107190 120811 100219  97675  89139  80978  79710  77215  62830  74893
## [11]  67529  64851  51895  43699  38429  42137  32888  40372  26307  32120
```

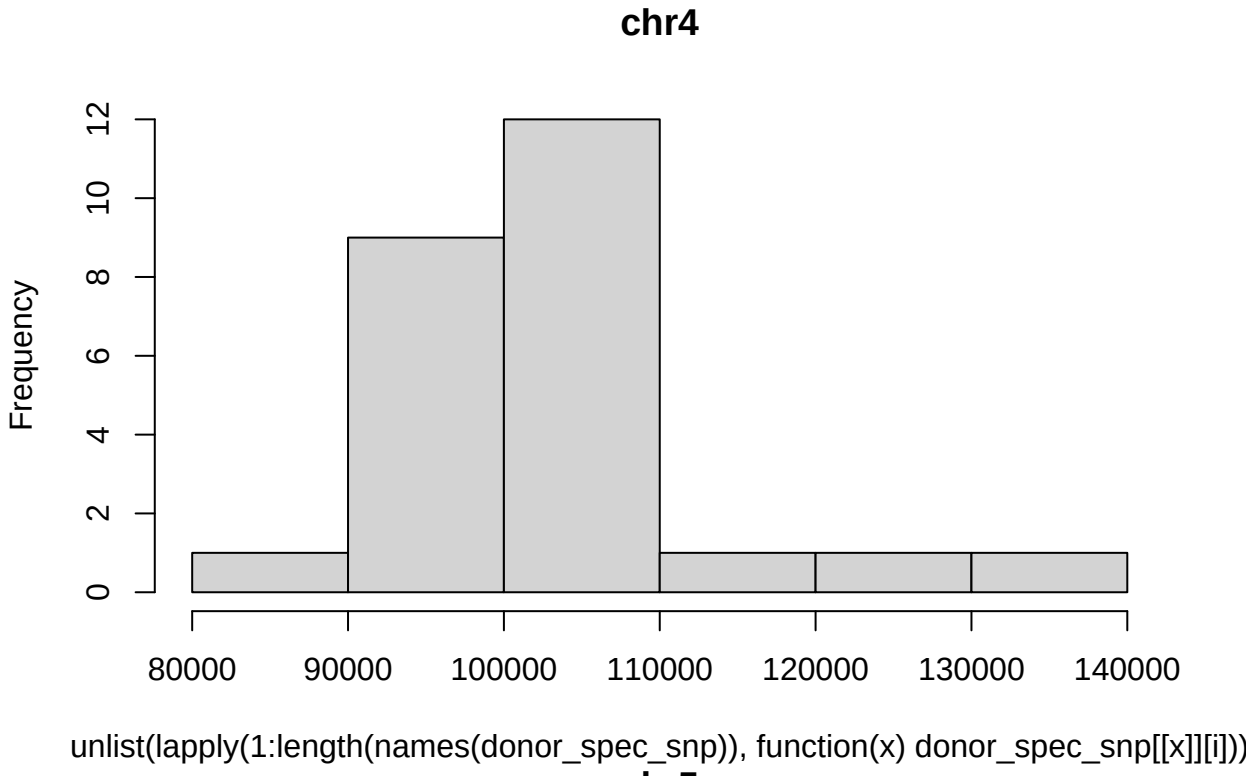
```
## [21]  19196  16737
##
## $pb2a
##  [1] 100082 107581  95675  92529  82532  79878  71041  69893  57843  67025
## [11]  62564  57173  49247  40335  35959  41166  31133  38873  24711  27736
## [21]  17891  13751
##
## $pb3a
##  [1] 110654 120897 108369  96594  90312  91067  77856  81512  62458  67671
## [11]  62465  65661  48416  40755  39940  40992  31456  39559  27002  32421
## [21]  19327  15712
##
## $pb4a
##  [1] 105352 113908 105425  96564  89643  86914  79793  74700  58061  69586
## [11]  67669  61833  53768  47360  35672  42147  34931  38960  26881  29555
## [21]  18666  16834
```

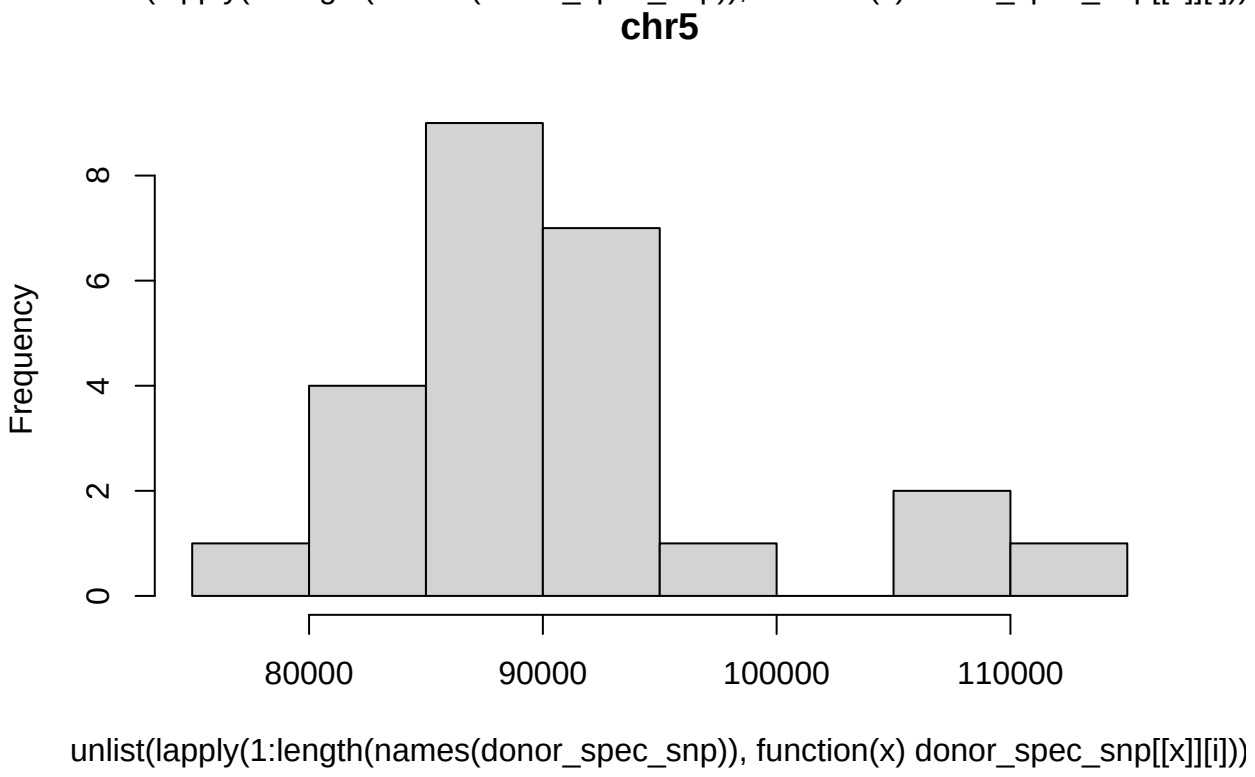
```
for (i in 1:22){
  hist(unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i])), main=paste0

}
```

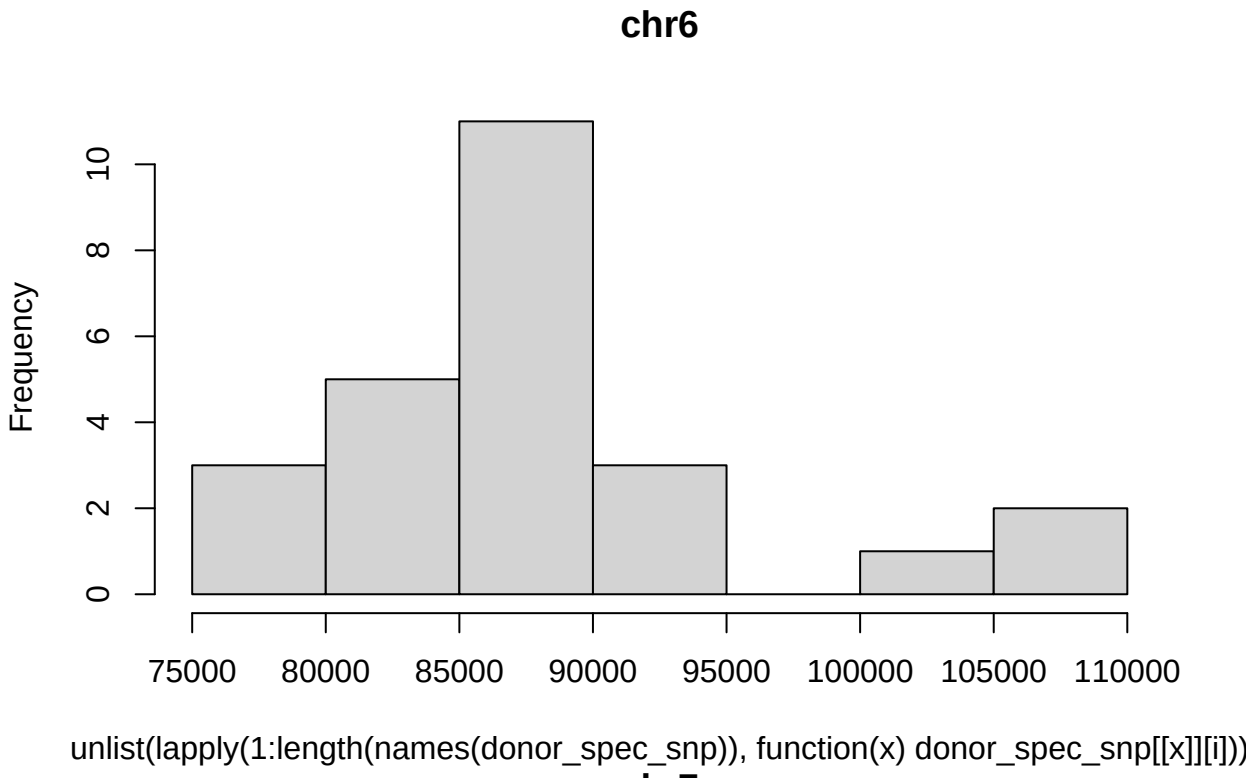


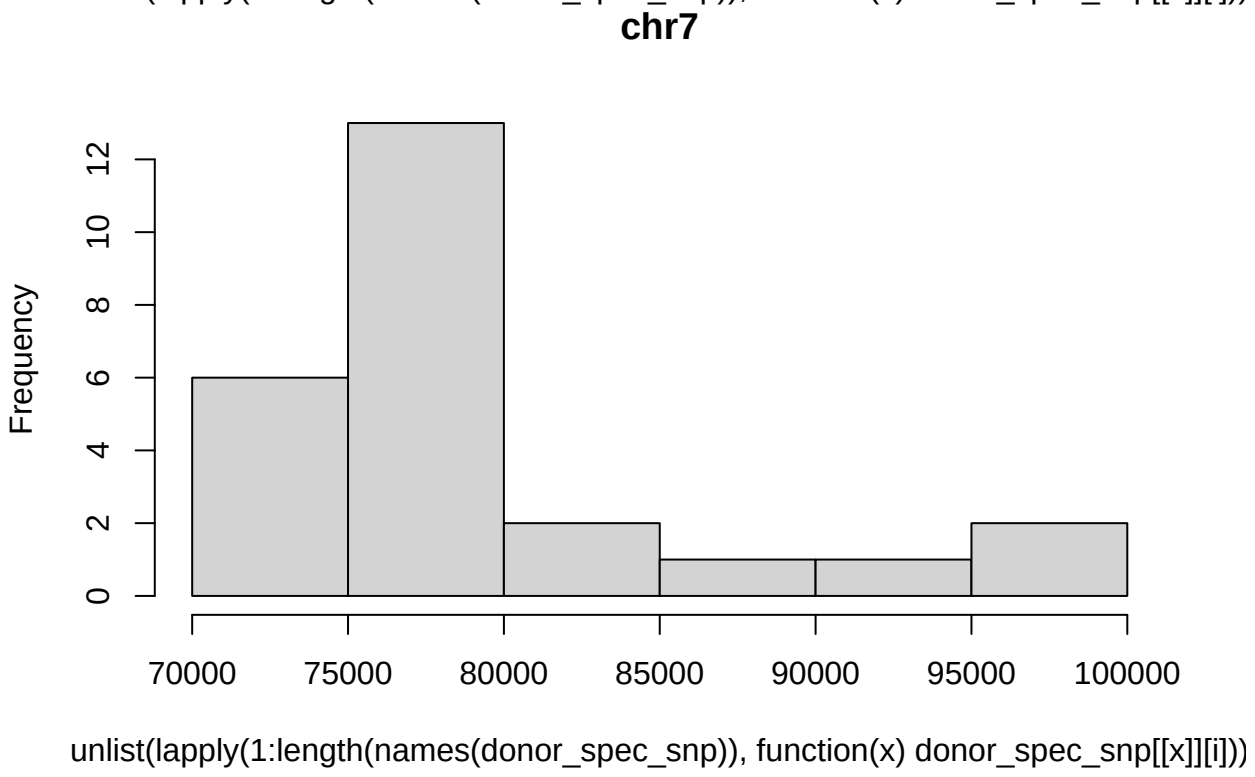

chr2
Frequency
0
2
4
6
8
10
100000
110000
120000
130000
140000
150000
160000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

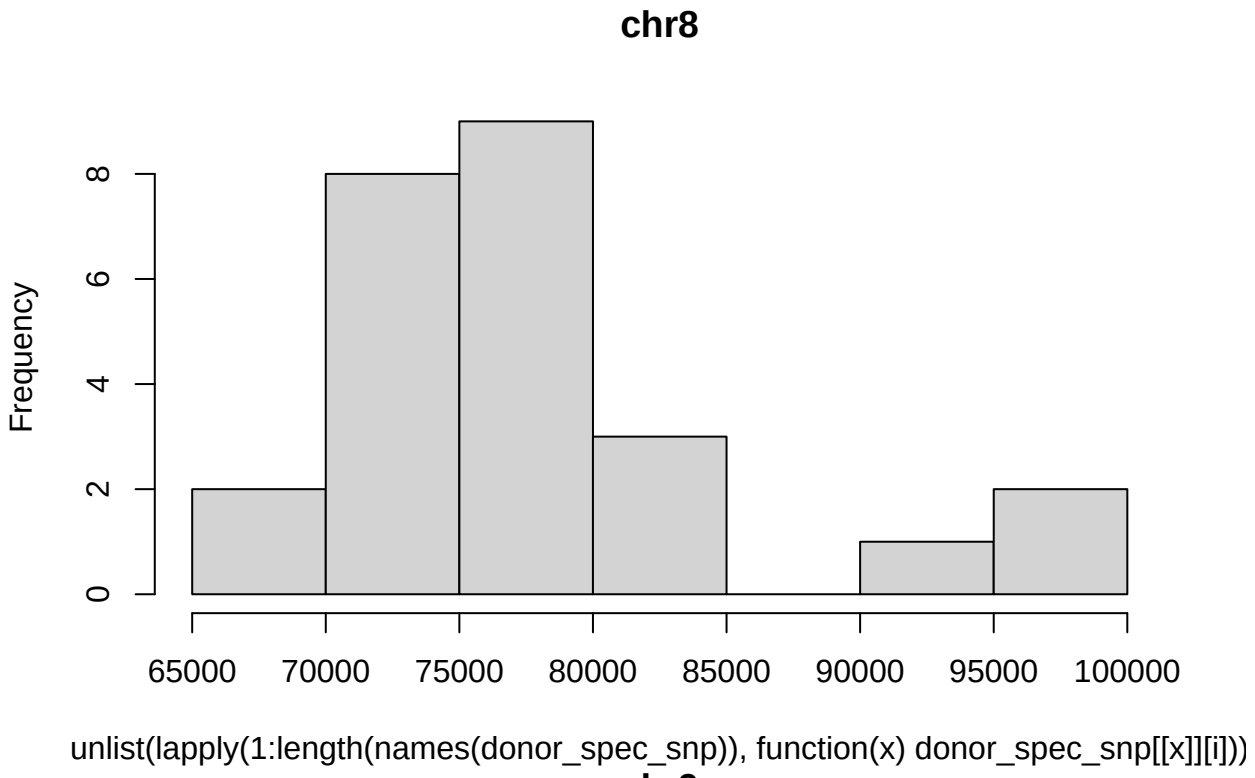

chr3
Frequency
0
2
4
6
8
10
90000
100000
110000
120000
130000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

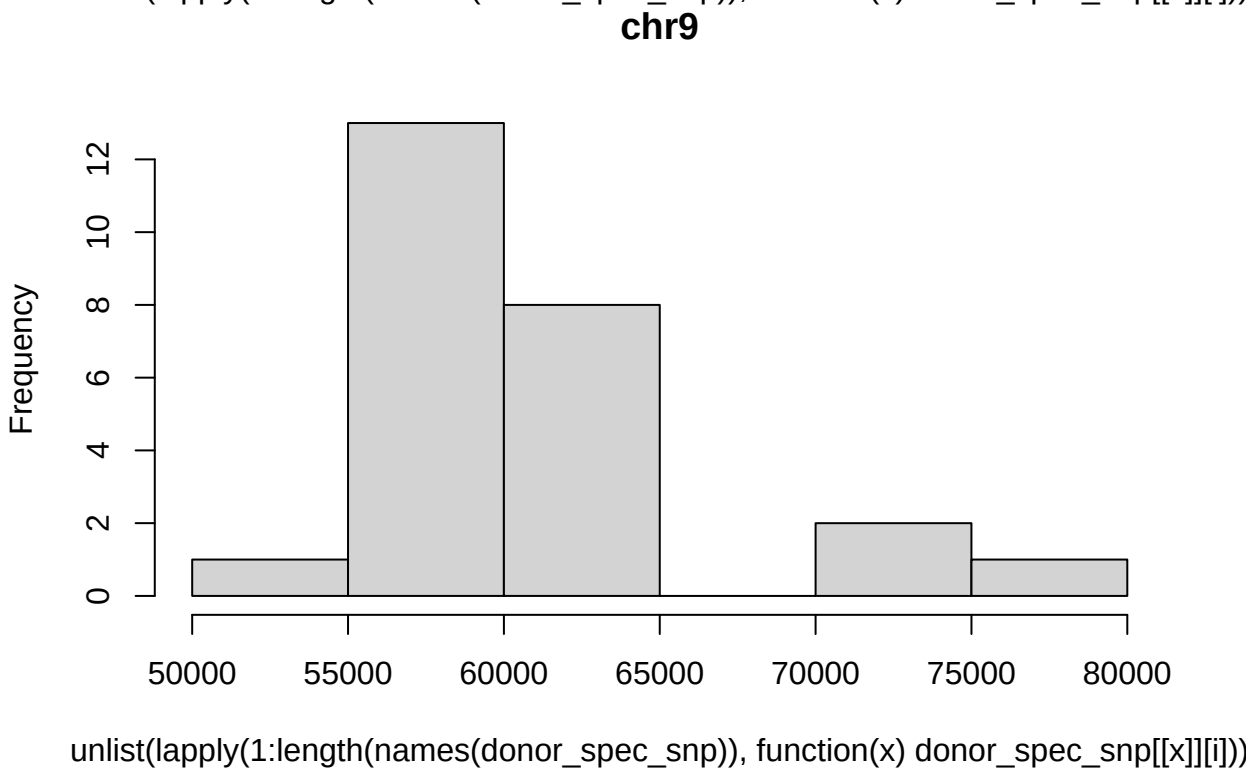

chr4
Frequency
0
2
4
6
8
10
12
80000
90000
100000
110000
120000
130000
140000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

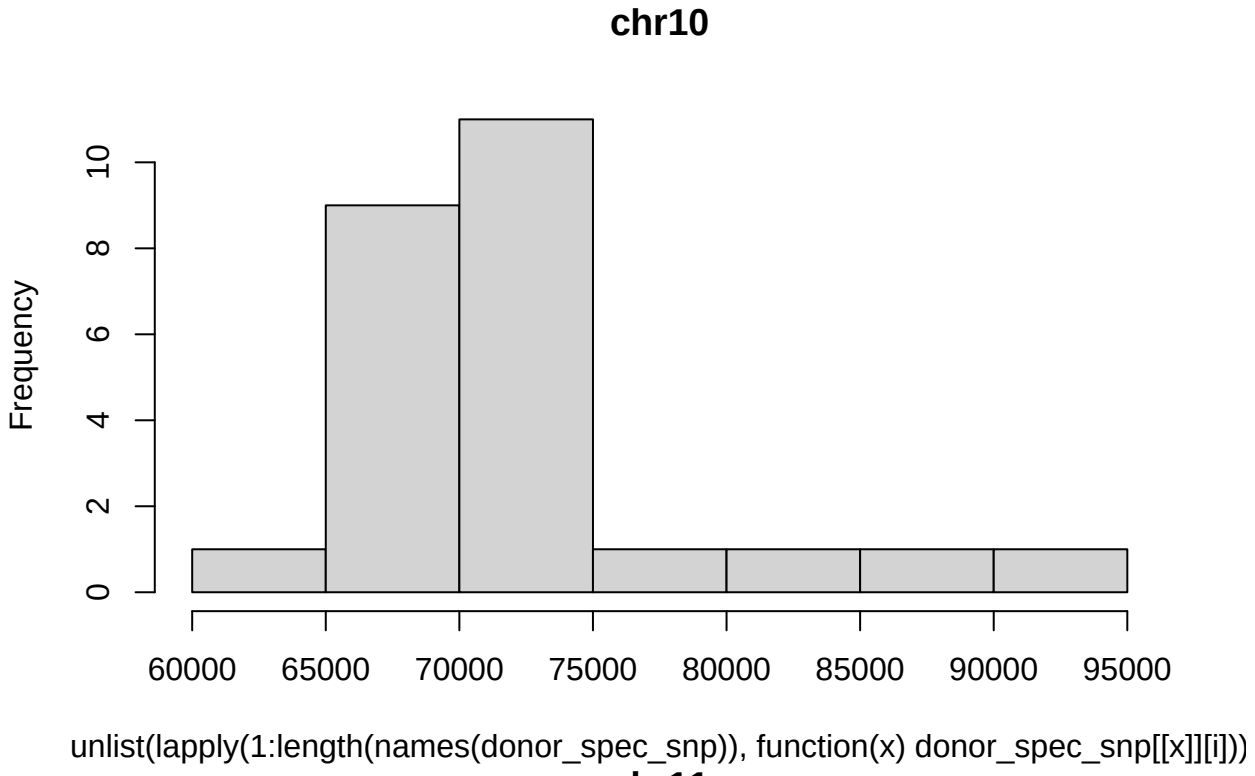

chr5
Frequency
0
2
4
6
8
80000
90000
100000
110000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

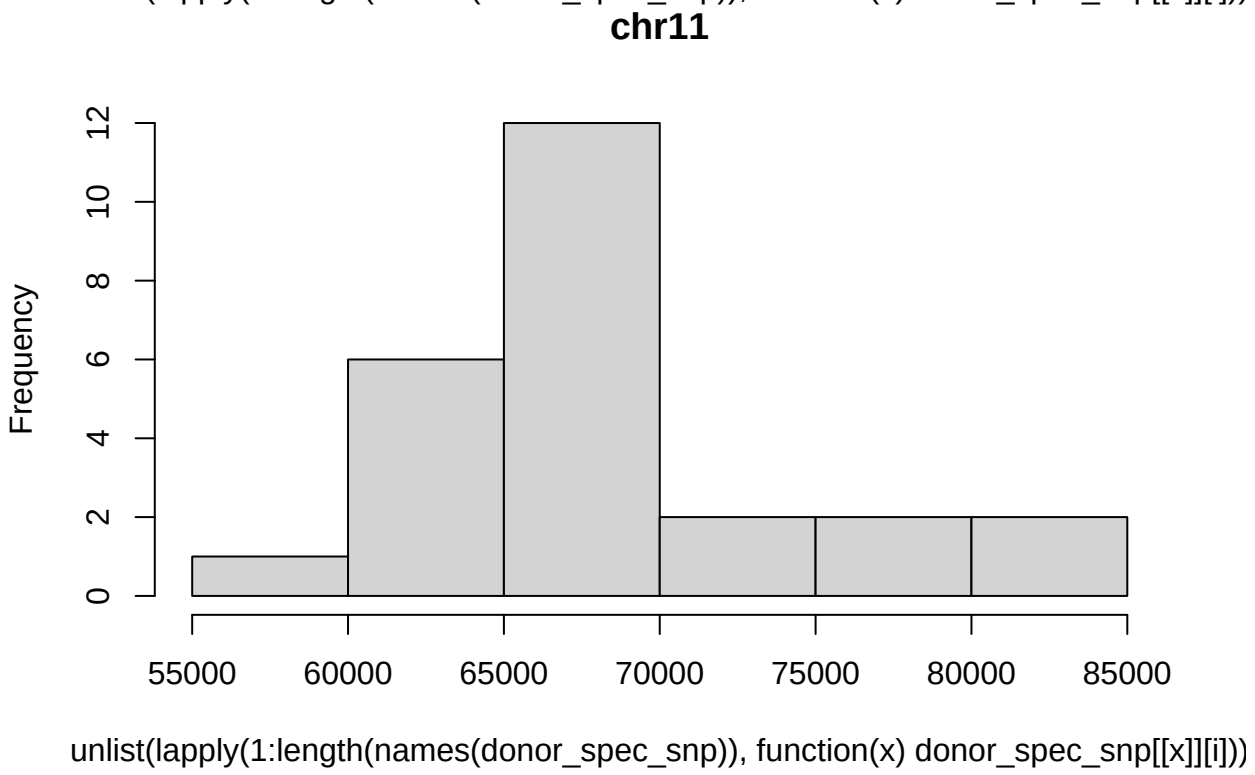

chr6
Frequency
0
2
4
6
8
10
75000
80000
85000
90000
95000
100000
105000
110000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

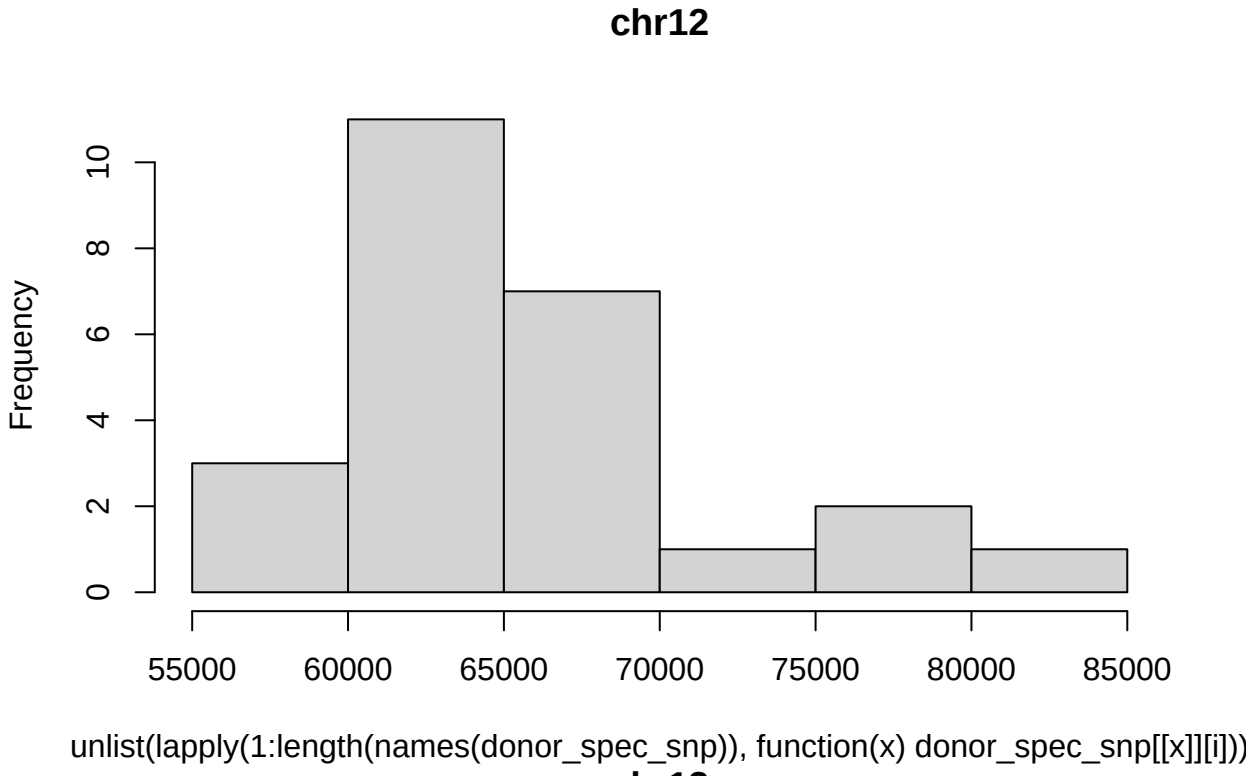

chr7
Frequency
0
2
4
6
8
10
12
70000
75000
80000
85000
90000
95000
100000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

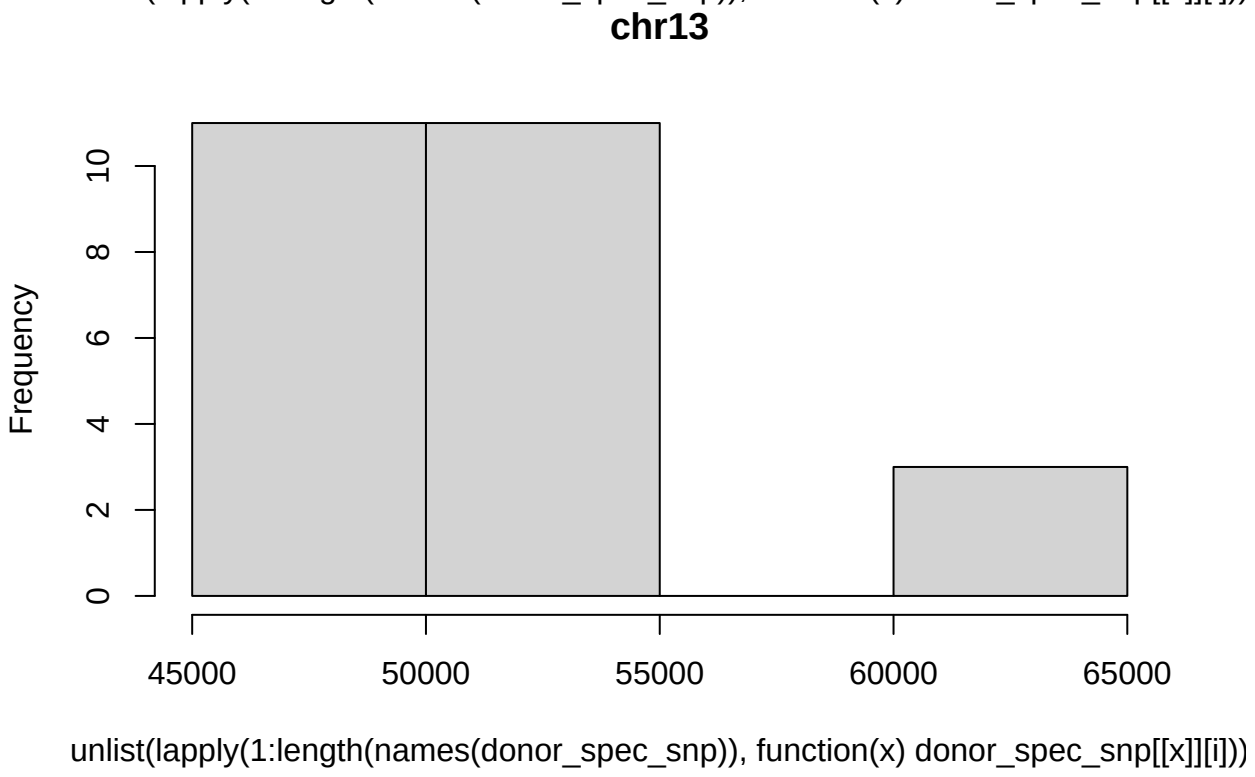

chr8
Frequency
0
2
4
6
8
65000
70000
75000
80000
85000
90000
95000
100000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

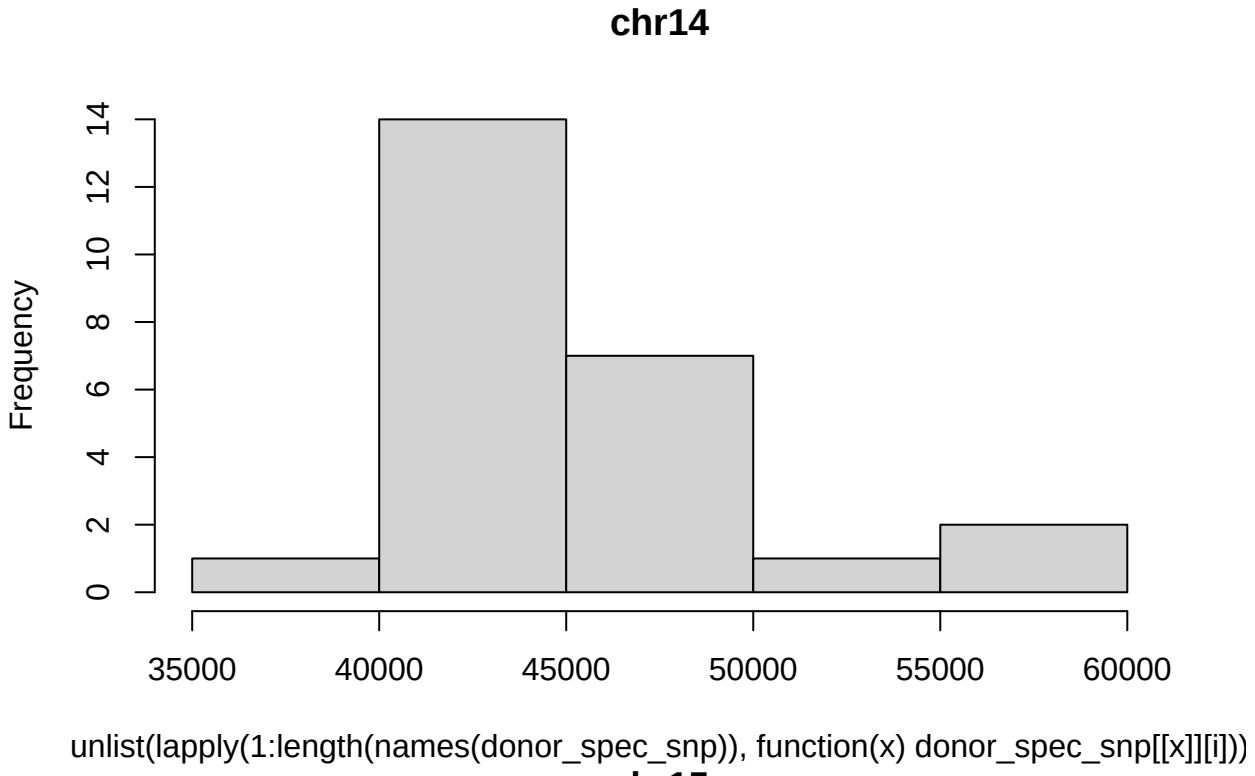

chr9
Frequency
0
2
4
6
8
10
12
50000
55000
60000
65000
70000
75000
80000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

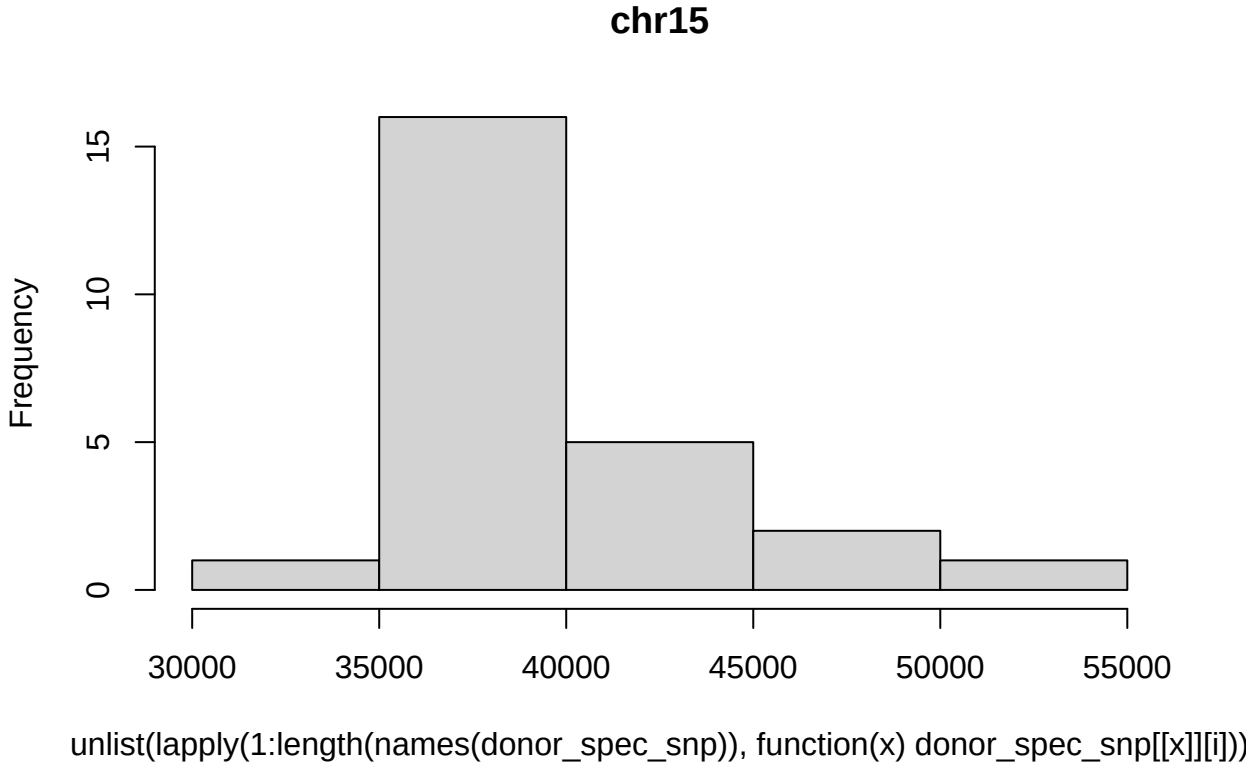

chr10
Frequency
0
2
4
6
8
10
60000
65000
70000
75000
80000
85000
90000
95000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

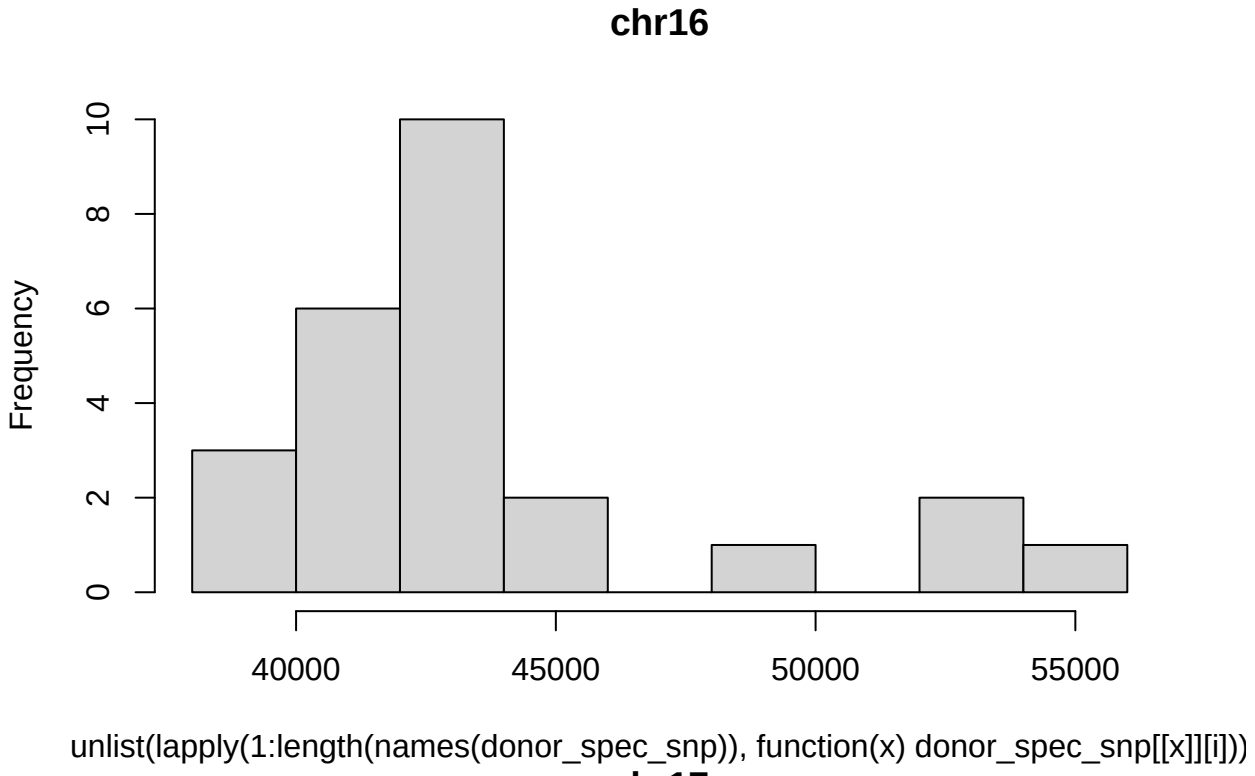

chr11
Frequency
0
2
4
6
8
10
12
55000
60000
65000
70000
75000
80000
85000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

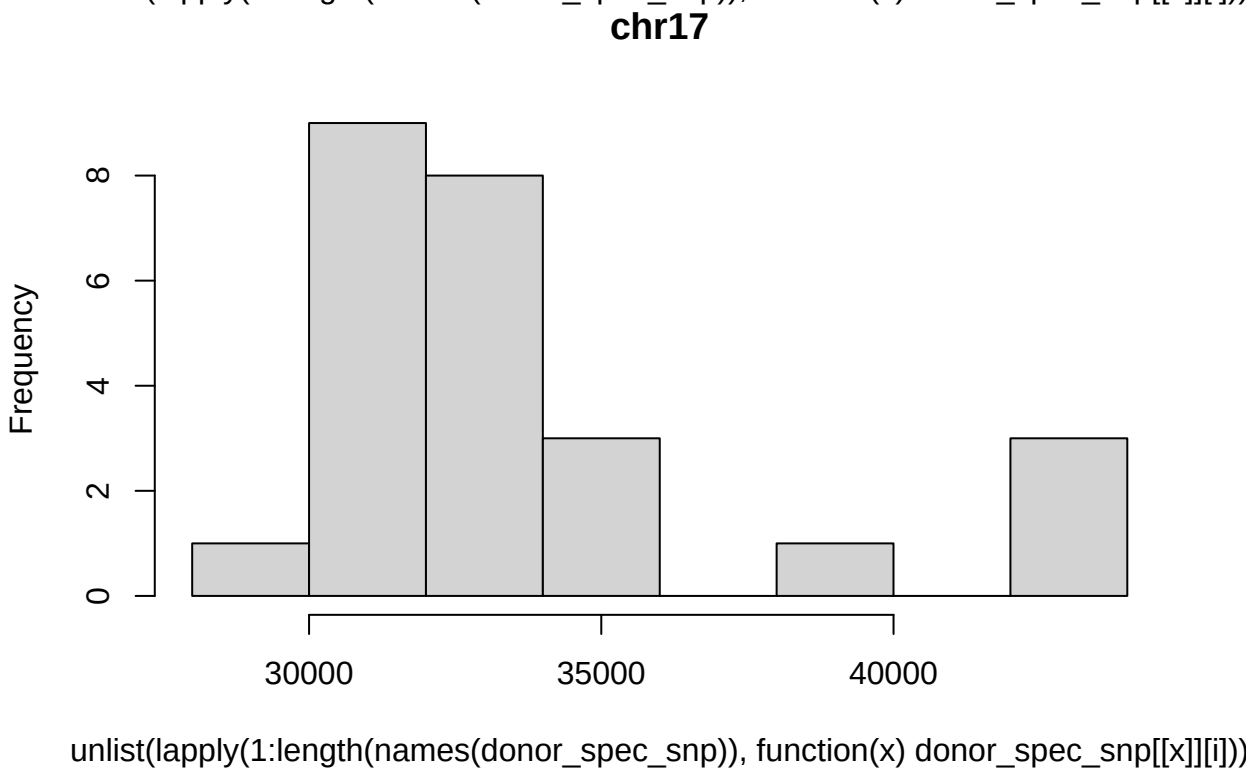

chr12
Frequency
0
2
4
6
8
10
55000
60000
65000
70000
75000
80000
85000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

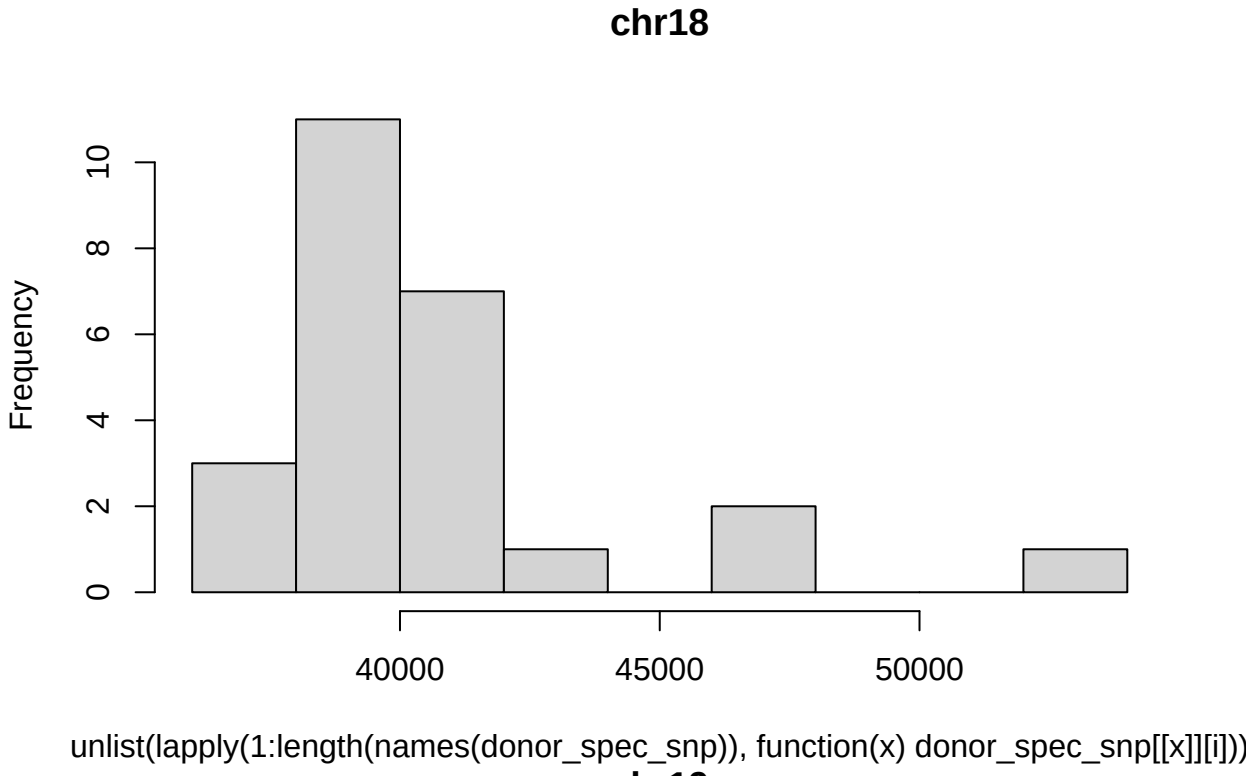

chr13
Frequency
0
2
4
6
8
10
45000
50000
55000
60000
65000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

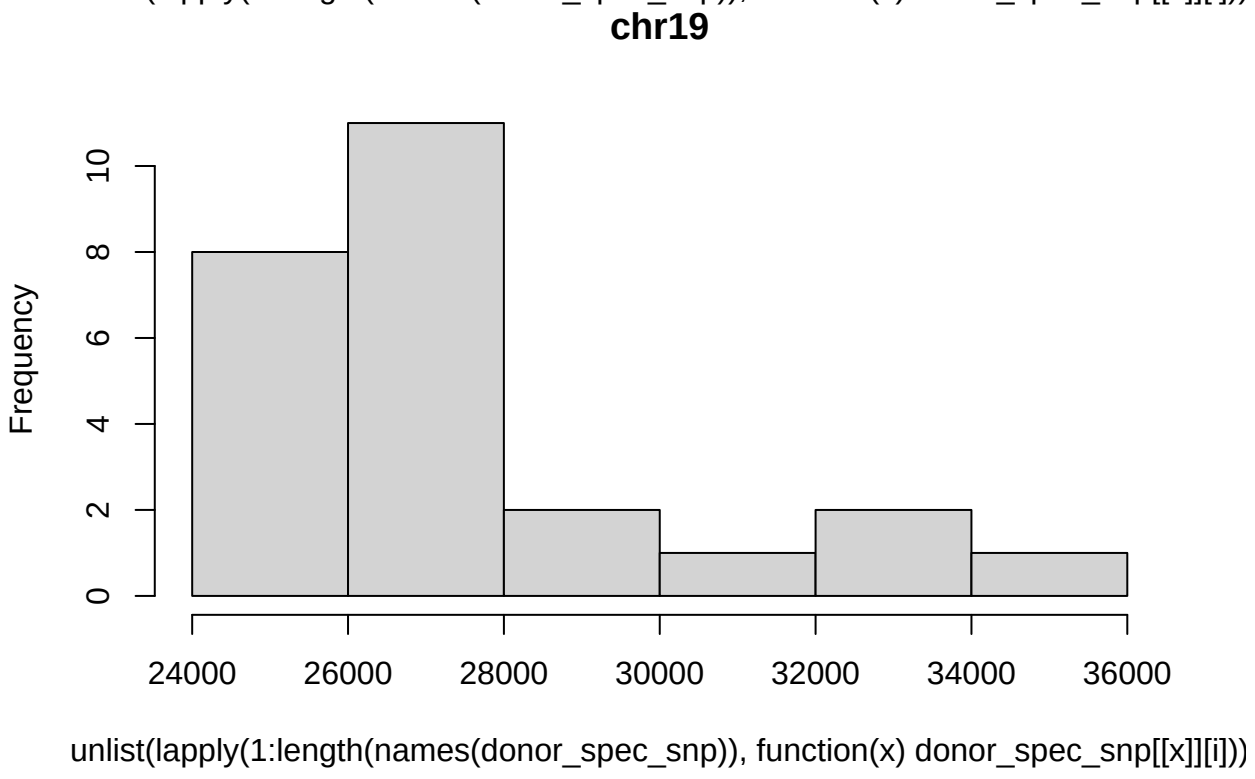

chr14
Frequency
0
2
4
6
8
10
12
14
35000
40000
45000
50000
55000
60000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

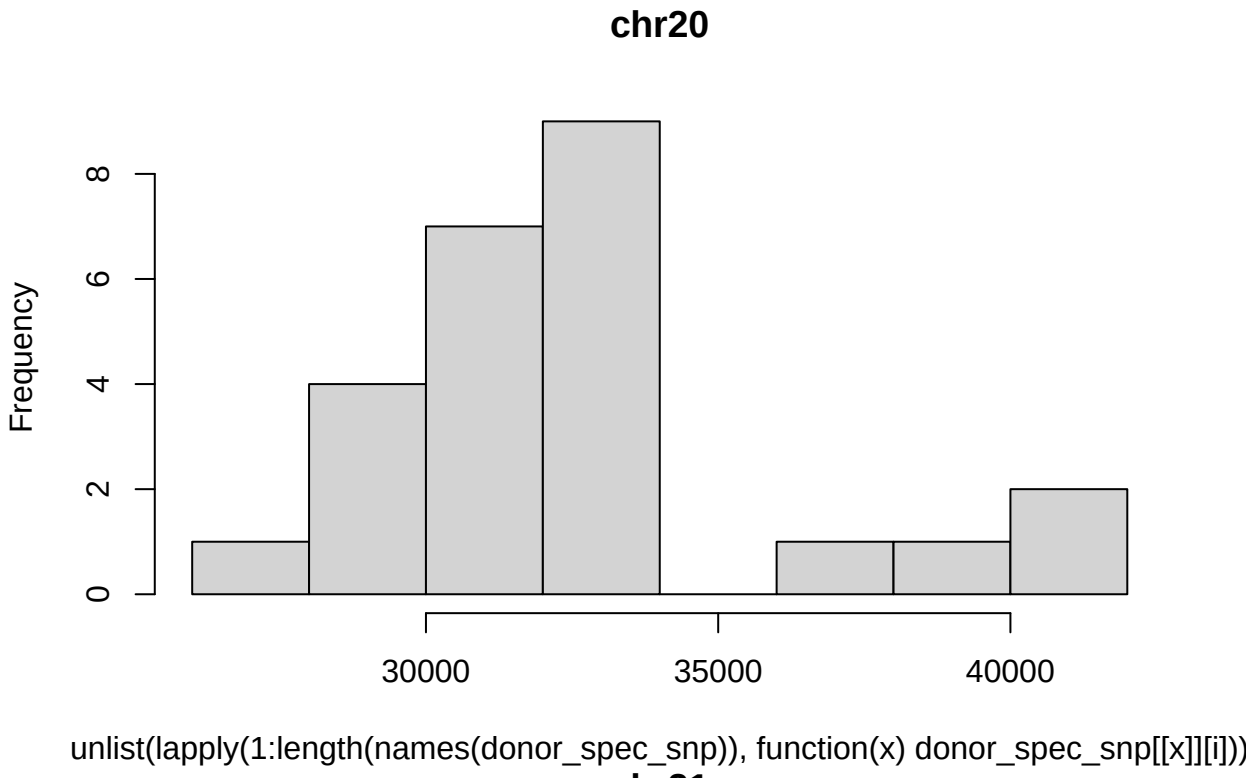

chr15
Frequency
0
5
10
15
30000
35000
40000
45000
50000
55000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

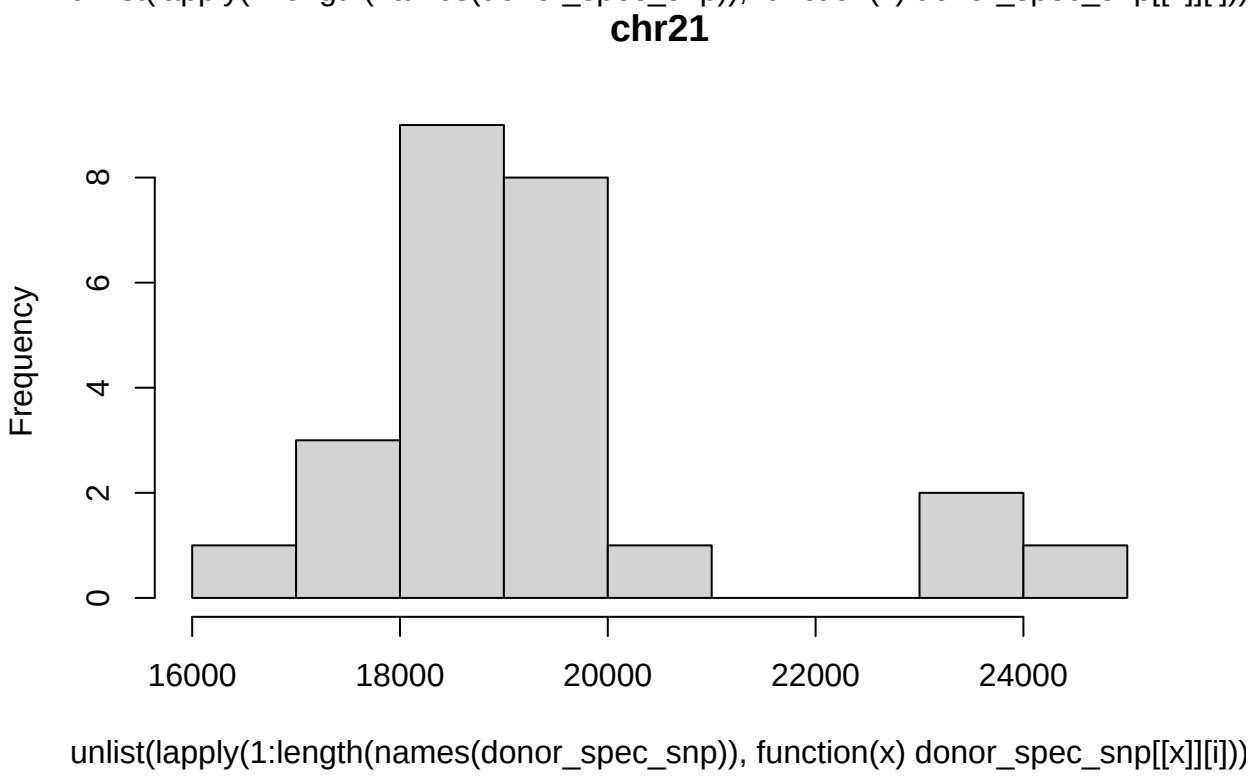

chr16
Frequency
0
2
4
6
8
10
40000
45000
50000
55000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))

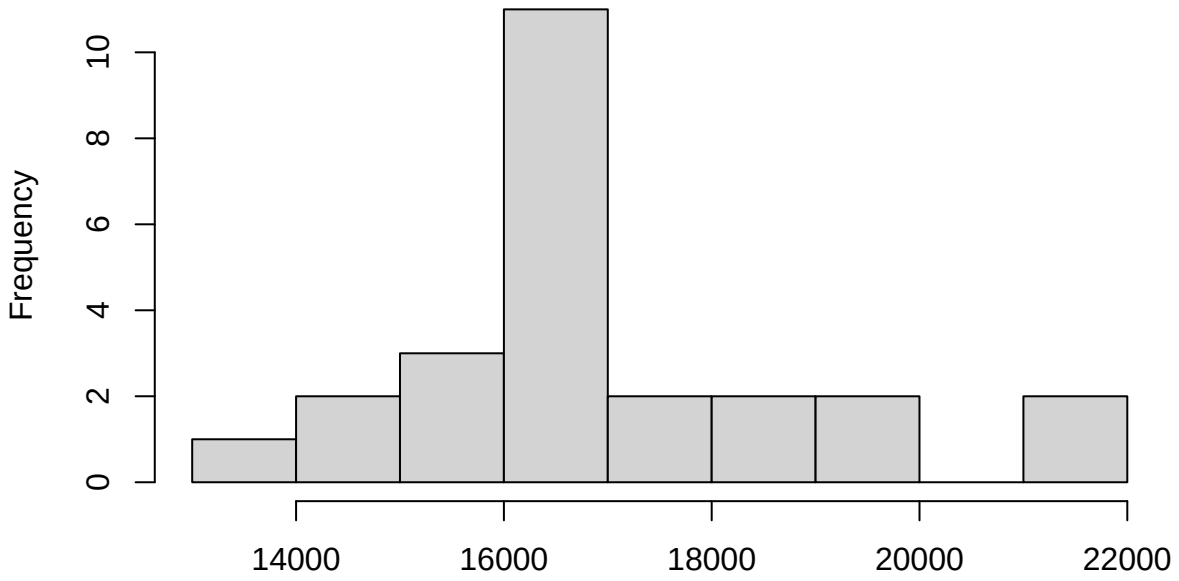

chr17
Frequency
0
2
4
6
8
30000
35000
40000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))


chr18
Frequency
0
2
4
6
8
10
40000
45000
50000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))


chr19
Frequency
0
2
4
6
8
10
24000
26000
28000
30000
32000
34000
36000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))


chr20
Frequency
0
2
4
6
8
30000
35000
40000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))


chr21
Frequency
0
2
4
6
8
16000
18000
20000
22000
24000
unlist(lapply(1:length(names(donor_spec_snp)), function(x) donor_spec_snp[[x]][i]))



these are chromosome specific counts of the number of hetSNPs following filtering.

Infertile Donors

```
min_inf_snp <- which(inf_df$num_snps == min(inf_df$num_snps))[1]
inf_df[min_inf_snp, c("sample", "chr", "num_snps")]
```

```
##     sample chr num_snps
## 548   pb2a  22    13751
```

```
max_inf_snp <- which(inf_df$num_snps == max(inf_df$num_snps))[1]
inf_df[max_inf_snp, c("sample", "chr", "num_snps")]
```

```
##    sample chr num_snps
## 49   pb3a   2   120897
```

```
quant_inf_snp <- quantile(inf_df$num_snps)
quant_inf_snp
```

```
##        0%       25%       50%       75%      100%
##  13751.00  37554.25  58655.00  81415.25 120897.00
```

Original Cohort of Donors

```
min_ninf_snp <- which(ninf_df$num_snps == min(ninf_df$num_snps))[1]
ninf_df[min_ninf_snp, c("sample", "chr", "num_snps")]
```

```
##     sample chr num_snps
## 531  nc8ab  22    14450
```

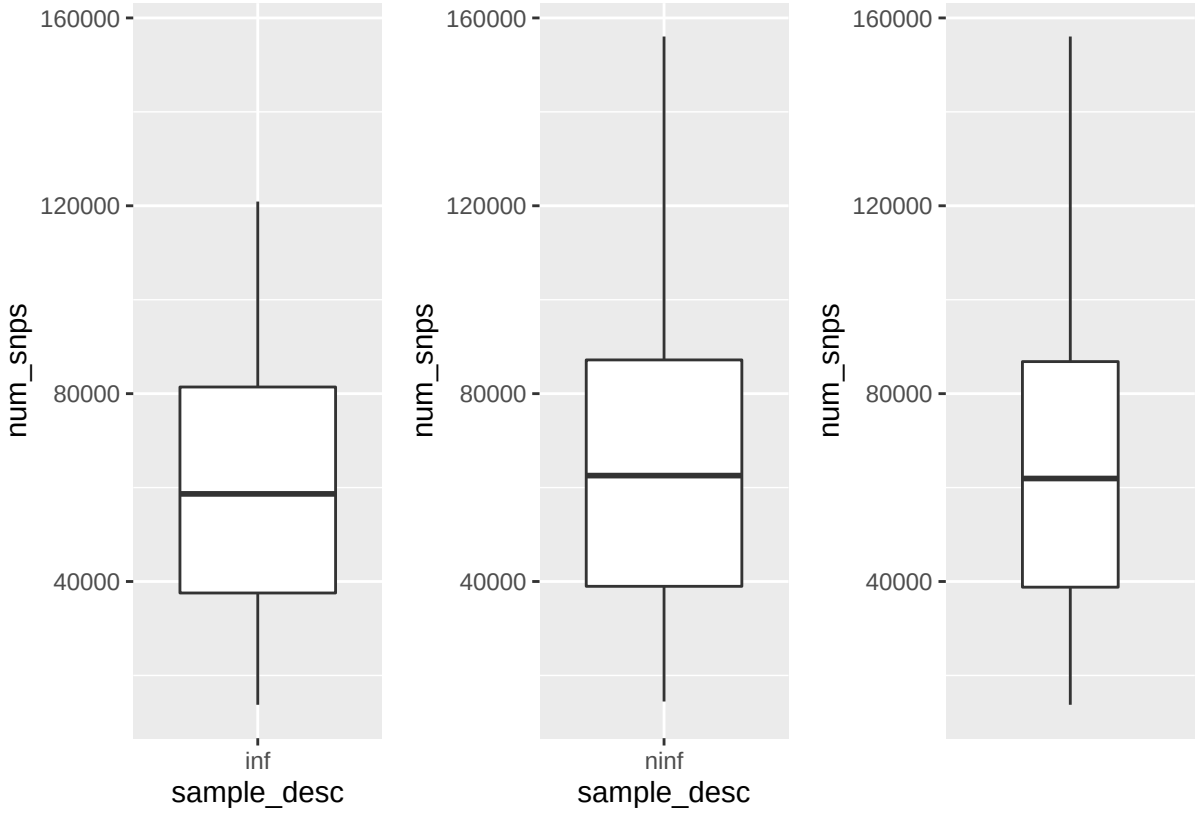
```
max_ninf_snp <- which(ninf_df$num_snps == max(ninf_df$num_snps))[1]
ninf_df[max_ninf_snp, c("sample", "chr", "num_snps")]
```

```
##      sample chr num_snps
## 43 nc25abcd   2   156043
```

```
quant_ninf_snp <- quantile(ninf_df$num_snps)
quant_ninf_snp
```

```
##        0%       25%       50%       75%      100%
##  14450.00  38974.50  62550.00  87182.25 156043.00
```

```
g_inf <- ggplot(inf_df, aes(x = sample_desc, y = num_snps)) + geom_boxplot() + scale_y_continuous(limits
g_ninf <- ggplot(ninf_df, aes(x = sample_desc, y = num_snps)) + geom_boxplot() + scale_y_continuous(limi
g_tot <- ggplot(full_df, aes(y = num_snps)) + geom_boxplot() + scale_y_continuous(limits=c(min(ninf_df$

g <- g_inf + g_ninf + g_tot
g
```



In this plot of the number of hetsNPs, after filtering, (donor and chromosome specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left. We seem to have under-estimated the average number of SNPs by using 30K for the simulations. But 30K is near the average for chr17 and chr20 (and is the standard deviation as seen below). And since we do better as we get more data, low-balling this should be no problem at all.

```
message(paste0("mean infertile donors: ", mean(inf_df$num_snps)))
```

```
## mean infertile donors: 60090.1636363636
```

```
message(paste0("sd infertile donors: ", sd(inf_df$num_snps)))
```

```
## sd infertile donors: 28758.3158049334
```

```
message(paste0("mean original donors: ", mean(ninf_df$num_snps)))
```

```
## mean original donors: 64007.5159090909
```

```
message(paste0("sd original donors: ", sd(ninf_df$num_snps)))
```

```
## sd original donors: 31031.4100456482
```

```
message(paste0('mean all: ', mean(full_df$num_snps)))
```

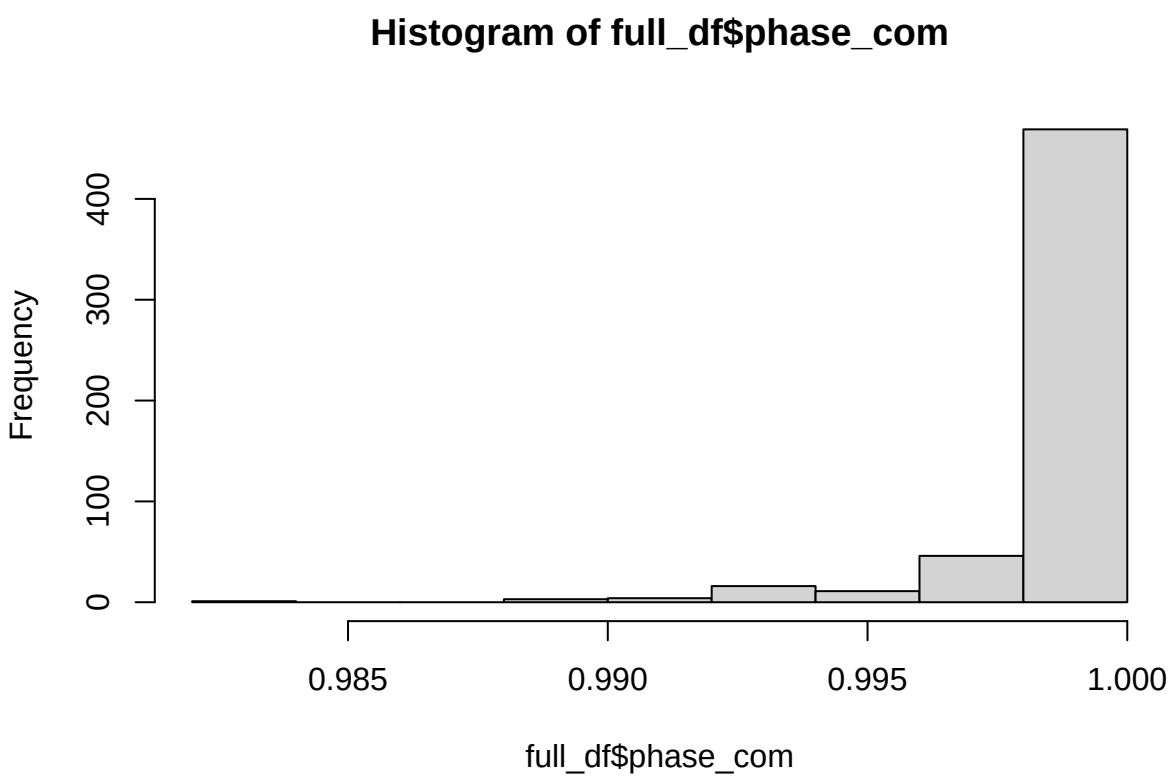
```
## mean all: 63224.0454545455
```

```
message(paste0("sd all: ", sd(full_df$num_snps)))
```

```
## sd all: 30605.0761941311
```

look at phasing completeness

```
hist(full_df$phase_com)
```




```
min(full_df$phase_com)
```

```
## [1] 0.9837623
```

```
max(full_df$phase_com)
```

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

```
mean(full_df$phase_com)
```

```
## [1] 0.9989478
```

```
sd(full_df$phase_com)
```

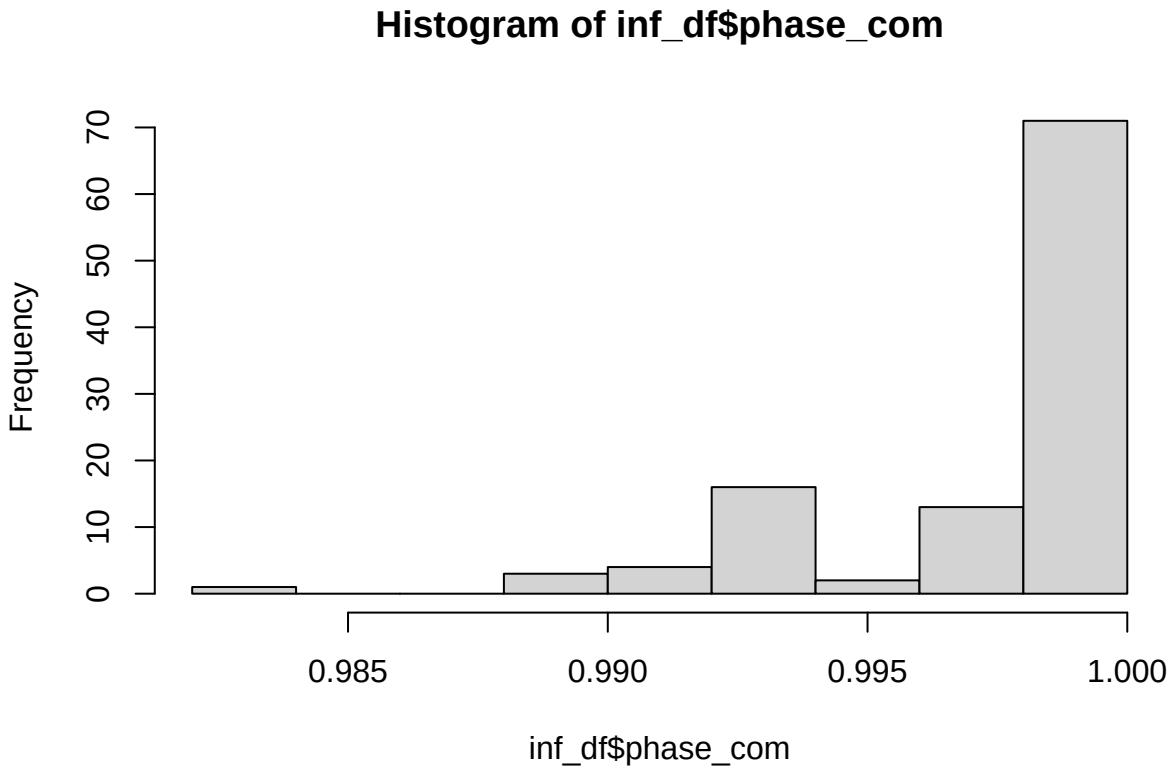
```
## [1] 0.001856551
```

```
quantile(full_df$phase_com)
```

```
##        0%       25%       50%       75%      100%
## 0.9837623 0.9992080 0.9995752 0.9998072 1.0000000
```

Infertile donors

```
hist(inf_df$phase_com)
```




```
min(inf_df$phase_com)
```

```
## [1] 0.9837623
```

```
max(inf_df$phase_com)
```

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

```
mean(inf_df$phase_com)
```

```
## [1] 0.9976857
```

```
sd(inf_df$phase_com)
```

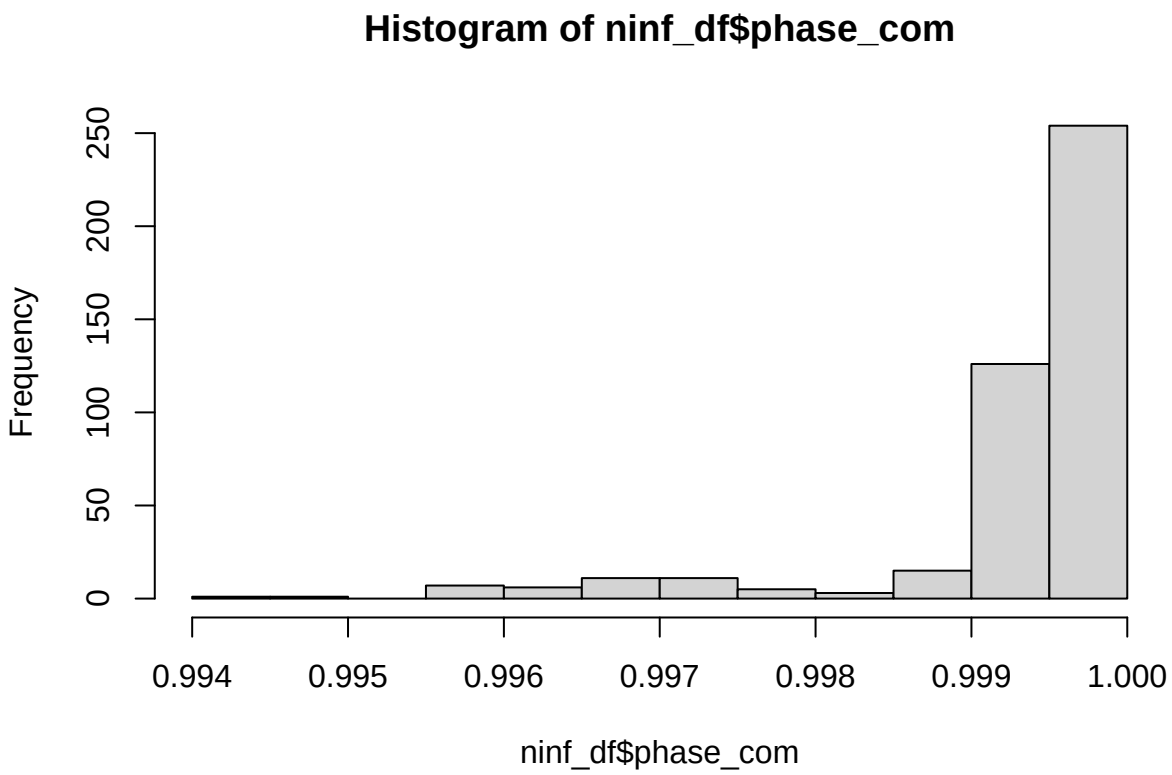
```
## [1] 0.003444241
```

```
quantile(inf_df$phase_com)
```

```
##        0%       25%       50%       75%      100%
## 0.9837623 0.9969388 0.9998653 1.0000000 1.0000000
```

Original set of donors

```
hist(ninf_df$phase_com)
```




```
min(ninf_df$phase_com)
```

```
## [1] 0.9943269
```

```
max(ninf_df$phase_com)
```

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

```
mean(ninf_df$phase_com)
```

```
## [1] 0.9992633
```

```
sd(ninf_df$phase_com)
```

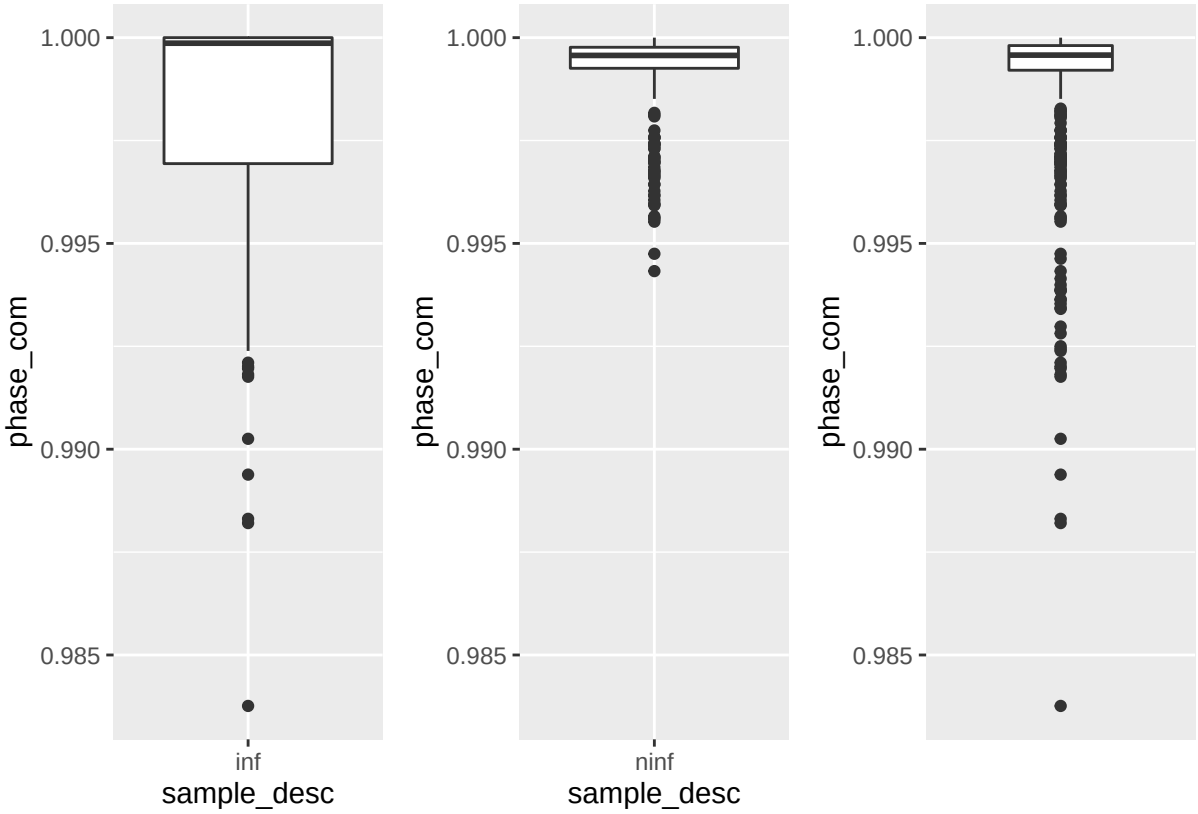
```
## [1] 0.0009306353
```

```
quantile(ninf_df$phase_com)
```

```
##        0%       25%       50%       75%      100%
## 0.9943269 0.9992557 0.9995677 0.9997658 1.0000000
```

```
min_inf_pcom <- which(inf_df$phase_com == min(inf_df$phase_com))[1]
min_ninf_pcom <- which(ninf_df$phase_com == min(ninf_df$phase_com))[1]
max_inf_pcom <- which(inf_df$phase_com == max(inf_df$phase_com))[1]
```

```
max_ninf_pcom <- which(ninf_df$phase_com == max(ninf_df$phase_com))[1]

g_inf <- ggplot(inf_df, aes(x = sample_desc, y = phase_com)) + geom_boxplot() + scale_y_continuous(limi
g_ninf <- ggplot(ninf_df, aes(x = sample_desc, y = phase_com)) + geom_boxplot() + scale_y_continuous(li
g_tot <- ggplot(full_df, aes(y = phase_com)) + geom_boxplot() + scale_y_continuous(limits=c(min(ninf_df

g <- g_inf + g_ninf + g_tot
g
```



In this plot of the phasing completeness (donor and chromosome specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left.

```
message(paste0("mean infertile donors: ", mean(inf_df$phase_com)))
```

```
## mean infertile donors: 0.997685651129779
```

```
message(paste0("sd infertile donors: ", sd(inf_df$phase_com)))
```

```
## sd infertile donors: 0.00344424081118475
```

```
message(paste0("mean original donors: ", mean(ninf_df$phase_com)))
```

```
## mean original donors: 0.999263293387455
```

```
message(paste0("sd original donors: ", sd(ninf_df$phase_com)))
```

```
## sd original donors: 0.000930635302046475
```

```
message(paste0('mean all: ', mean(full_df$phase_com)))
```

```
## mean all: 0.99894776493592
```

```
message(paste0("sd all: ", sd(full_df$phase_com)))
```

```
## sd all: 0.00185655080249439
```

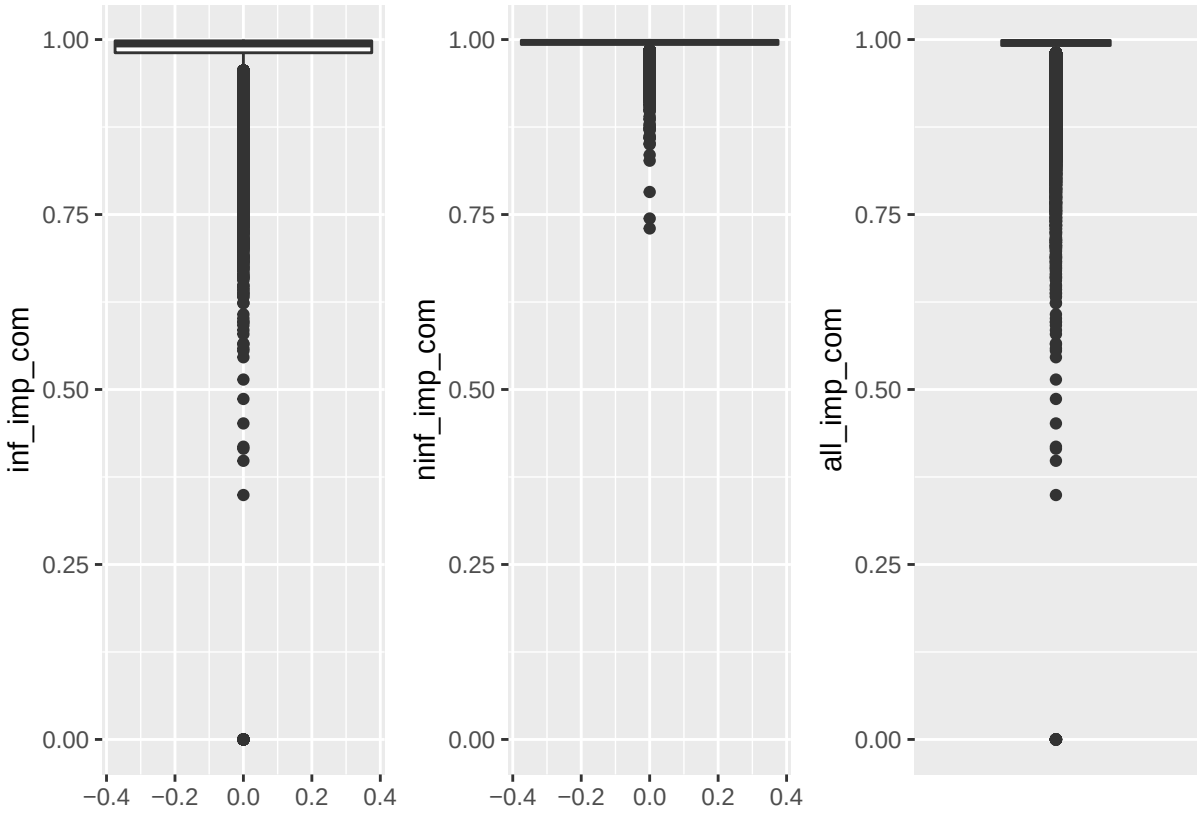
Infertile donors appear to have slightly suppressed phasing completeness, but overall comparable to the original donors. The mean for both cohorts together is 99.89 +- 0.19% complete

look at gamete imputation completeness

```
undo_str_vec <- function(str_vec){
  undone <- as.numeric(unlist(strsplit(str_vec, '_')))
  return(undone)
}
```

```
all_imp_com <- unlist(lapply(1:nrow(full_df), function(x) undo_str_vec(full_df$imp_com[x])))
inf_imp_com <- unlist(lapply(1:nrow(inf_df), function(x) undo_str_vec(inf_df$imp_com[x])))
ninf_imp_com <- unlist(lapply(1:nrow(ninf_df), function(x) undo_str_vec(ninf_df$imp_com[x])))
```

```
min_inf_icom <- which(inf_imp_com == min(inf_imp_com))[1]
min_ninf_icom <- which(ninf_imp_com == min(ninf_imp_com))[1]
max_inf_icom <- which(inf_imp_com == max(inf_imp_com))[1]
max_ninf_icom <- which(ninf_imp_com == max(ninf_imp_com))[1]

g_inf <- ggplot(as.data.table(inf_imp_com), aes(y=inf_imp_com)) + geom_boxplot() + scale_y_continuous(li
g_ninf <- ggplot(as.data.table(ninf_imp_com), aes(y=ninf_imp_com)) + geom_boxplot() + scale_y_continuous
g_tot <- ggplot(as.data.table(all_imp_com), aes(y=all_imp_com)) + geom_boxplot() + scale_y_continuous(li

g <- g_inf + g_ninf + g_tot
g
```



In this plot of the gamete imputation completeness (donor, chromosome, and gamete specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left. Note that the min for the infertile donors is 0% completeness, compared to about 73% for the original donor set. Thankfully, this is a major outlier as the 25th percentile is around 98% (as seen below).

```
message(paste0("mean infertile donors: ", mean(inf_imp_com)))
```

```
## mean infertile donors: 0.984493441292405
```

```
message(paste0("sd infertile donors: ", sd(inf_imp_com)))
```

```
## sd infertile donors: 0.0252344306971294
```

```
message(paste0("mean original donors: ", mean(ninf_imp_com)))
```

```
## mean original donors: 0.994895770267944
```

```
message(paste0("sd original donors: ", sd(ninf_imp_com)))
```

```
## sd original donors: 0.00563185767020054
```

```
message(paste0('mean all: ', mean(all_imp_com)))
```

```
## mean all: 0.992081468797671
```

```
message(paste0("sd all: ", sd(all_imp_com)))
```

```
## sd all: 0.0147230476642437
```

Original set of donors

```
min(ninf_imp_com)
```

```
## [1] 0.7301333
```

```
max(ninf_imp_com)
```

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

```
mean(ninf_imp_com)
```

```
## [1] 0.9948958
```

```
sd(ninf_imp_com)
```

```
## [1] 0.005631858
```

```
quantile(ninf_imp_com)
```

```
##        0%       25%       50%       75%      100%
## 0.7301333 0.9929673 0.9964488 0.9989216 1.0000000
```

Infertile donors

```
min(inf_imp_com)
```

```
## [1] 0
```

```
max(inf_imp_com)
```

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

```
mean(inf_imp_com)
```

```
## [1] 0.9844934
```

```
sd(inf_imp_com)
```

```
## [1] 0.02523443
```

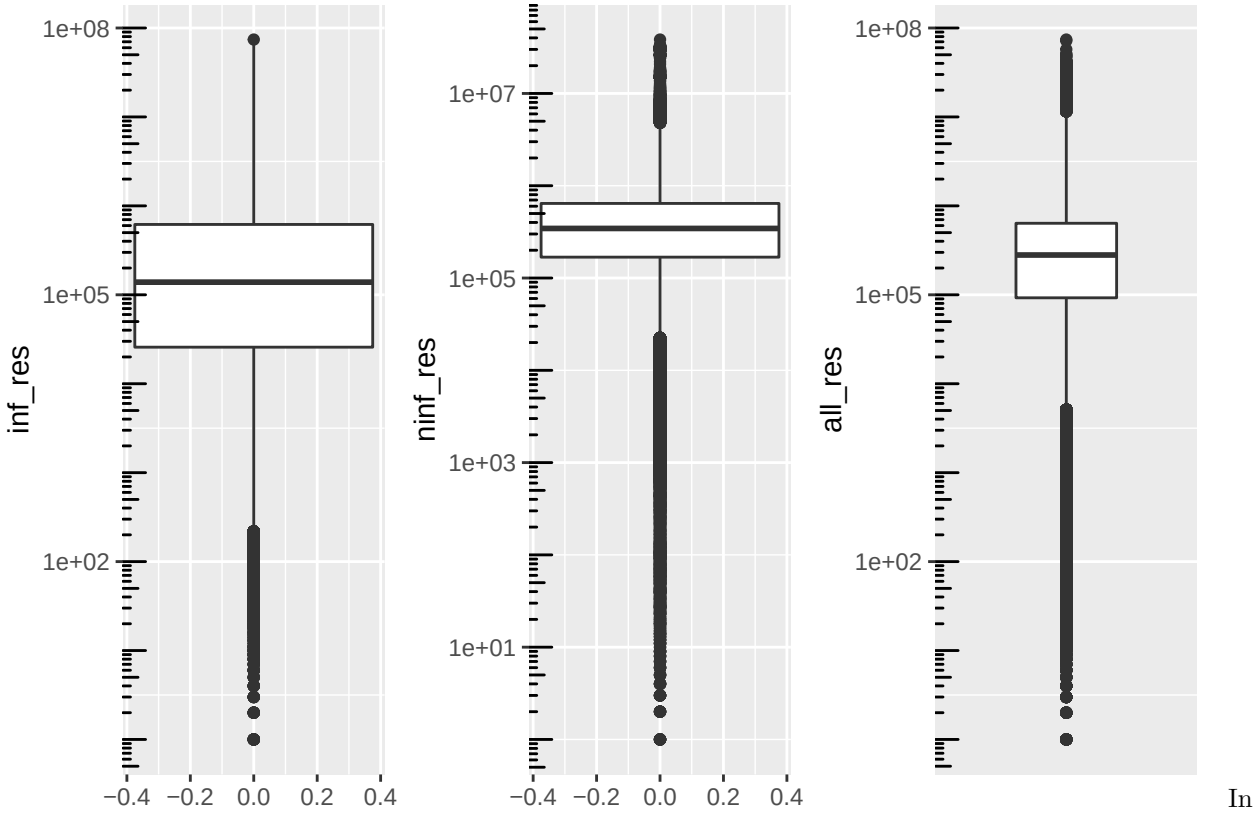
```
quantile(inf_imp_com)
```

```
##        0%       25%       50%       75%      100%
## 0.0000000 0.9811093 0.9928138 0.9981267 1.0000000
```

breakpoint resolution (in base pairs for real this time!)

```
all_res <- unlist(lapply(1:nrow(full_df), function(x) undo_str_vec(full_df$res[x])))
inf_res <- unlist(lapply(1:nrow(inf_df), function(x) undo_str_vec(inf_df$res[x])))
ninf_res <- unlist(lapply(1:nrow(ninf_df), function(x) undo_str_vec(ninf_df$res[x])))
```

```
min_inf_res <- which(inf_res == min(inf_res))[1]
min_ninf_res <- which(ninf_res == min(ninf_res))[1]
max_inf_res <- which(inf_res == max(inf_res))[1]
max_ninf_res <- which(ninf_res == max(ninf_res))[1]

g_inf <- ggplot(as.data.table(inf_res), aes(y=inf_res)) + geom_boxplot() + scale_y_log10() + annotation_
g_ninf <- ggplot(as.data.table(ninf_res), aes(y=ninf_res)) + geom_boxplot() + scale_y_log10() + annotati
g_tot <- ggplot(as.data.table(all_res), aes(y=all_res)) + geom_boxplot() + scale_x_discrete(labels=c("bo

g <- g_inf + g_ninf + g_tot
g
```



In this plot of the predicted meiotic breakpoint resolution (donor, chromosome, and gamete specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left. Note that the breakpoint resolution is actual base pairs this time!

```
message(paste0("mean infertile donors: ", mean(inf_res)))
```

```
## mean infertile donors: 681981.29779972
```

```
message(paste0("sd infertile donors: ", sd(inf_res)))
```

```
## sd infertile donors: 1632818.47693665
```

```
message(paste0("mean original donors: ", mean(ninf_res)))
```

```
## mean original donors: 525489.221135424
```

```
message(paste0("sd original donors: ", sd(ninf_res)))
```

```
## sd original donors: 891172.565087
```

```
message(paste0('mean all: ', mean(all_res)))
```

```
## mean all: 589046.205488826
```

```
message(paste0("sd all: ", sd(all_res)))
```

```
## sd all: 1249136.47041477
```

Original set of donors

```
min(ninf_res)
```

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

```
max(ninf_res)
```

```
## [1] 38379432
```

```
mean(ninf_res)
```

```
## [1] 525489.2
```

```
sd(ninf_res)
```

```
## [1] 891172.6
```

```
quantile(ninf_res)
```

```
##         0%        25%        50%        75%       100%
##        1.0   168386.8   344533.5   643101.0 38379432.0
```

Infertile donors

```
min(inf_res)
```

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

```
max(inf_res)
```

```
## [1] 74158225
```

```
mean(inf_res)
```

```
## [1] 681981.3
```

```
sd(inf_res)
```

```
## [1] 1632818
```

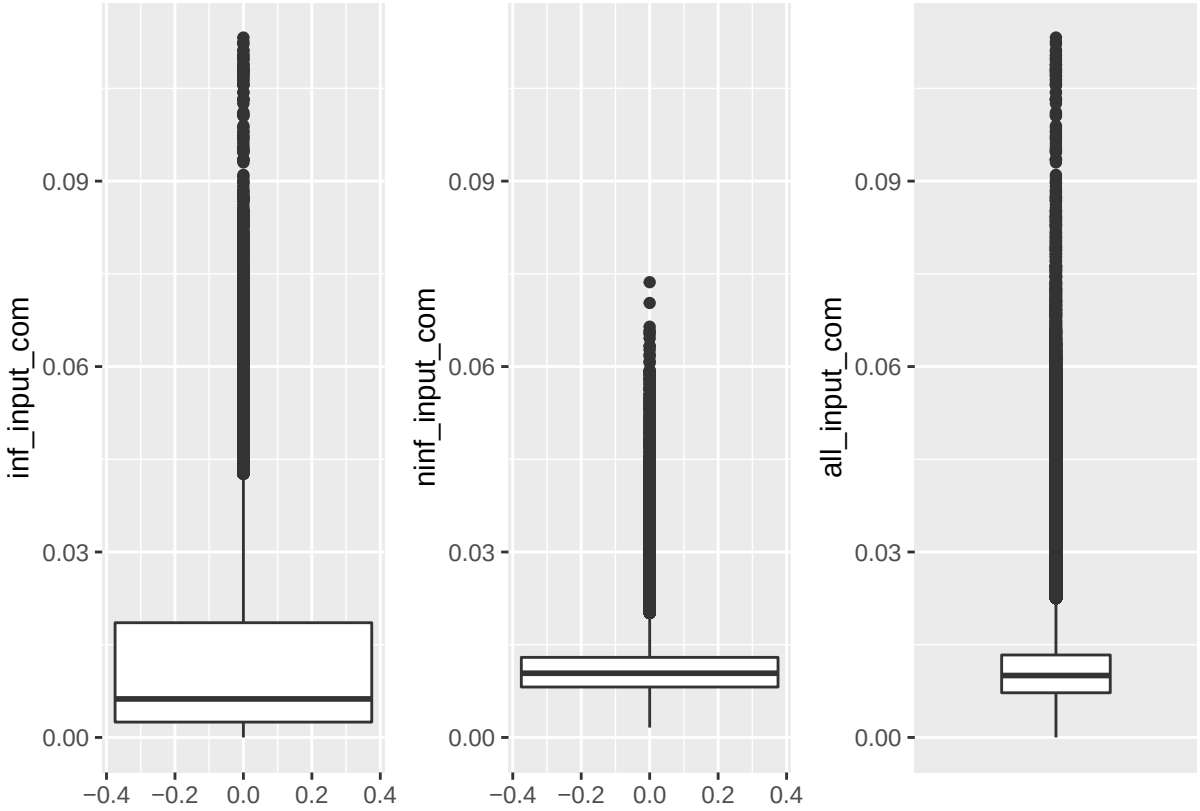
```
quantile(inf_res)
```

```
##       0%      25%      50%      75%     100%
##        1    25767   138748   617862 74158225
```

completeness of input data

```
all_input_com <- unlist(lapply(1:nrow(full_df), function(x) undo_str_vec(full_df$input_com[x])))
inf_input_com <- unlist(lapply(1:nrow(inf_df), function(x) undo_str_vec(inf_df$input_com[x])))
ninf_input_com <- unlist(lapply(1:nrow(ninf_df), function(x) undo_str_vec(ninf_df$input_com[x])))
```

```
min_inf_inpcom <- which(inf_input_com == min(inf_input_com))[1]
min_ninf_inpcom <- which(ninf_input_com == min(ninf_input_com))[1]
max_inf_inpcom <- which(inf_input_com == max(inf_input_com))[1]
max_ninf_inpcom <- which(ninf_input_com == max(ninf_input_com))[1]

g_inf <- ggplot(as.data.table(inf_input_com), aes(y=inf_input_com)) + geom_boxplot() + scale_y_continuou
g_ninf <- ggplot(as.data.table(ninf_input_com), aes(y=ninf_input_com)) + geom_boxplot() + scale_y_conti
g_tot <- ggplot(as.data.table(all_input_com), aes(y=all_input_com)) + geom_boxplot() + scale_y_continuou

g <- g_inf + g_ninf + g_tot
g
```



In this plot of the input completeness (donor, chromosome, and gamete specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left. Note that the max for the infertile donors is >9% completeness, compared to about 7.5% for the original donor set.

```
message(paste0("mean infertile donors: ", mean(inf_input_com)))
```

```
## mean infertile donors: 0.0125899577231895
```

```
message(paste0("sd infertile donors: ", sd(inf_input_com)))
```

```
## sd infertile donors: 0.0137877513816203
```

```
message(paste0("mean original donors: ", mean(ninf_input_com)))
```

```
## mean original donors: 0.0109367623112843
```

```
message(paste0("sd original donors: ", sd(ninf_input_com)))
```

```
## sd original donors: 0.0039034538230503
```

```
message(paste0('mean all: ', mean(all_input_com)))
```

```
## mean all: 0.0113840266007828
```

```
message(paste0("sd all: ", sd(all_input_com)))
```

```
## sd all: 0.00794261390235215
```

Original set of donors

```
min(ninf_input_com)
```

```
## [1] 0.00160222
```

```
max(ninf_input_com)
```

```
## [1] 0.07364775
```

```
mean(ninf_input_com)
```

```
## [1] 0.01093676
```

```
sd(ninf_input_com)
```

```
## [1] 0.003903454
```

```
quantile(ninf_input_com)
```

```
##          0%         25%         50%         75%        100%
## 0.001602220 0.008160725 0.010391712 0.012956522 0.073647754
```

Infertile donors

```
min(inf_input_com)
```

```
## [1] 0
```

```
max(inf_input_com)
```

```
## [1] 0.1132229
```

```
mean(inf_input_com)
```

```
## [1] 0.01258996
```

```
sd(inf_input_com)
```

```
## [1] 0.01378775
```

```
quantile(inf_input_com)
```

```
##          0%         25%         50%         75%        100%
## 0.000000000 0.002485154 0.006229436 0.018559119 0.113222933
```

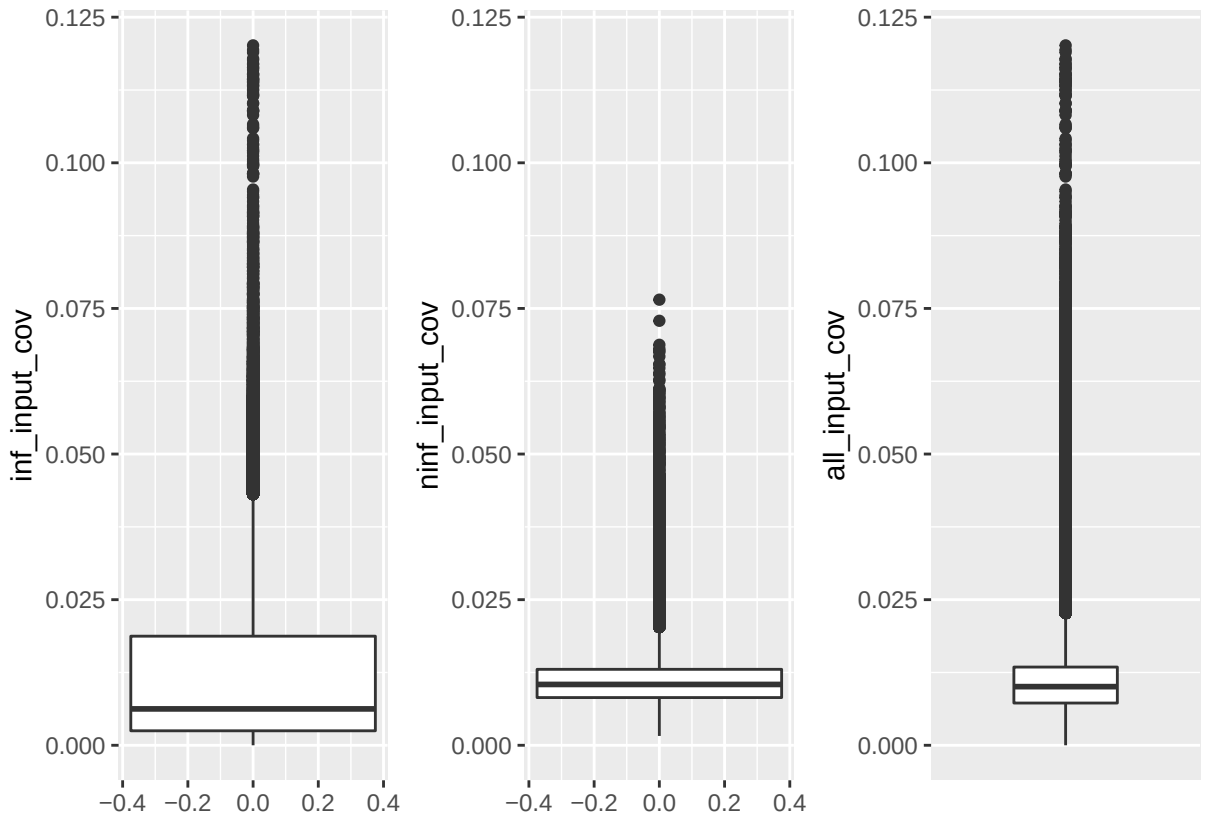
Completeness to coverage

for this next group of EDA, we'll translate to the missing genotype rate (MGR) and then translate missing genotype rate to coverage. I assume that MGR is the opposite of completeness or the # of NAs / number of SNPs (as opposed to the number of non-NAs / number of SNPs which is completeness). So this is just subtracting the completeness from 1. we then translate it to coverage by taking the -log.

```
all_input_cov <- unlist(lapply(1:nrow(full_df), function(x) -log(1-undo_str_vec(full_df$input_com[x]))))
inf_input_cov <- unlist(lapply(1:nrow(inf_df), function(x) -log(1-undo_str_vec(inf_df$input_com[x]))))
ninf_input_cov <- unlist(lapply(1:nrow(ninf_df), function(x) -log(1-undo_str_vec(ninf_df$input_com[x]))))
```

```
min_inf_inpcov <- which(inf_input_cov == min(inf_input_cov))[1]
min_ninf_inpcov <- which(ninf_input_cov == min(ninf_input_cov))[1]
max_inf_inpcov <- which(inf_input_cov == max(inf_input_cov))[1]
max_ninf_inpcov <- which(ninf_input_cov == max(ninf_input_cov))[1]

g_inf <- ggplot(as.data.table(inf_input_cov), aes(y=inf_input_cov)) + geom_boxplot() + scale_y_continuou
g_ninf <- ggplot(as.data.table(ninf_input_cov), aes(y=ninf_input_cov)) + geom_boxplot() + scale_y_conti
g_tot <- ggplot(as.data.table(all_input_cov), aes(y=all_input_cov)) + geom_boxplot() + scale_y_continuou
```

```
g <- g_inf + g_ninf + g_tot
g
```



In this plot of the input approximate coverage (donor, chromosome, and gamete specific), the infertile donors are on the left, the original donor set is in the middle, the combined set is on the left.

```
message(paste0("mean infertile donors: ", mean(inf_input_cov)))
```

```
## mean infertile donors: 0.012768770888996
```

```
message(paste0("sd infertile donors: ", sd(inf_input_cov)))
```

```
## sd infertile donors: 0.0141137318356838
```

```
message(paste0("mean original donors: ", mean(ninf_input_cov)))
```

```
## mean original donors: 0.0110048316029012
```

```
message(paste0("sd original donors: ", sd(ninf_input_cov)))
```

```
## sd original donors: 0.00396008182515815
```

```
message(paste0('mean all: ', mean(all_input_cov)))
```

```
## mean all: 0.0114820571298003
```

```
message(paste0("sd all: ", sd(all_input_cov)))
```

```
## sd all: 0.00812067213491068
```

Original set of donors

```
min(ninf_input_cov)
```

```
## [1] 0.001603505
```

```
max(ninf_input_cov)
```

```
## [1] 0.07650072
```

```
mean(ninf_input_cov)
```

```
## [1] 0.01100483
```

```
sd(ninf_input_cov)
```

```
## [1] 0.003960082
```

```
quantile(ninf_input_cov)
```

```
##          0%         25%         50%         75%        100%
## 0.001603505 0.008194206 0.010446083 0.013041190 0.076500721
```

Infertile donors

```
min(inf_input_cov)
```

```
## [1] 0
```

```
max(inf_input_cov)
```

```
## [1] 0.1201617
```

```
mean(inf_input_cov)
```

```
## [1] 0.01276877
```

```
sd(inf_input_cov)
```

```
## [1] 0.01411373
```

```
quantile(inf_input_cov)
```

```
##          0%         25%         50%         75%        100%
## 0.000000000 0.002488247 0.006248920 0.018733501 0.120161662
```