SNV examination

Nadzeya Boyeva

2024-11-01

Imports

Import VCF files: gen_vcf1 – VCF retrieved from genomic data, sample 1; gen_vcf2 – VCF retrieved from genomic data, sample 2; gen_merged_vcf – genomic data, sample 1 and 2 merged on alignment step; nas_vcf1 – nascent RNA data, sample 1; nas_vcf2 – nascent RNA data, sample 2; nas_vcf3 – nascent RNA data, sample 3; nas_vcf_merged – nascent RNA data, sample 1, 2 and 3 merged after mapping step.

```
fd <- '/home/nadzeya/praktika/'
gen_vcf1_path = file.path(fd, "gen1_snps_f.vcf.gz")
gen_vcf2_path = file.path(fd, "gen2_snps_f.vcf.gz")
gen_merged_vcf_path <- file.path(fd, "gen_merged_snps_f.vcf.gz")
nas_vcf1_path <- file.path(fd, "nas1_snps_f.vcf.gz")
nas_vcf2_path <- file.path(fd, "nas2_snps_f.vcf.gz")
nas_vcf3_path <- file.path(fd, "nas3_2_snps_f.vcf.gz")
nas_merged_vcf_path <- file.path(fd, "nas_merged_snps_f.vcf.gz")

filepaths <- c(gen_vcf1_path,
               gen_vcf2_path,
               gen_merged_vcf_path,
               nas_vcf1_path,
               nas_vcf2_path,
               nas_vcf3_path,
               nas_merged_vcf_path)

labels <- c('Gen 1',
            'Gen 2',
            'Gen Merged',
            'Nas 1',
            'Nas 2',
            'Nas 3',
            'Nas Merged')
```

Comparison of SNV quality metrics across files

Get the dataframe of QC metrics available and their descriptions.

```
gen1 <- readVcf(gen_vcf1_path, "hg38")
```

```
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
```

```r
descriptions <- info(header(gen1))
descriptions_df <- data.frame(Description = descriptions$Description,
                              row.names = rownames(descriptions))
descriptions_df
```

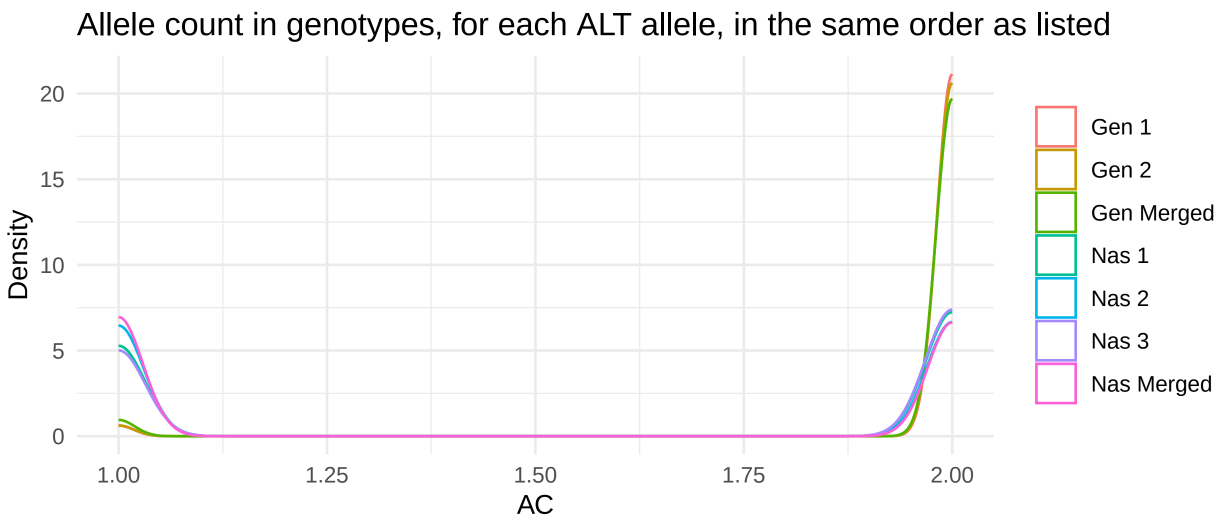
```
##
## AC                                                                                       Allele coun
## AF                                                                                                Al
## AN
## BaseQRankSum                                                                                       Z-
## DP
## ExcessHet
## FS                                                                                              Phre
## InbreedingCoeff                    Inbreeding coefficient as estimated from the genotype likelihoods
## MLEAC              Maximum likelihood expectation (MLE) for the allele counts (not necessarily the sa
## MLEAF           Maximum likelihood expectation (MLE) for the allele frequency (not necessarily the sa
## MQ
## MQRankSum                                                                                  Z-score F
## QD
## ReadPosRankSum                                                                                 Z-scor
## SOR                                                                                              Sym
```

Create information dataframes with QC metrics for all VCF files and plot these metrics densities.

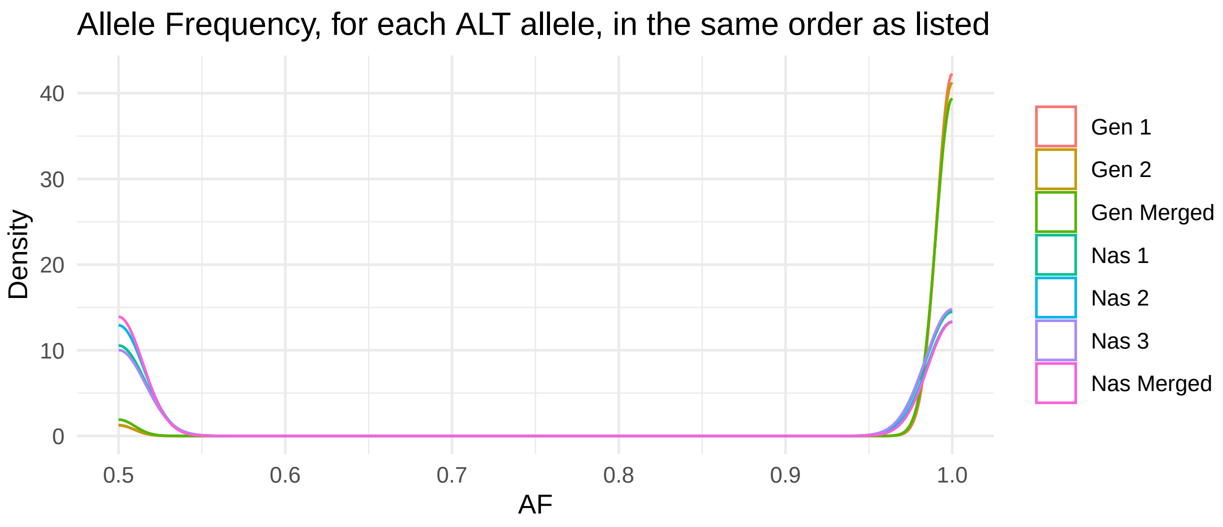
```r
full_info_table <- create_full_info_table(filepaths)
```

```
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
```

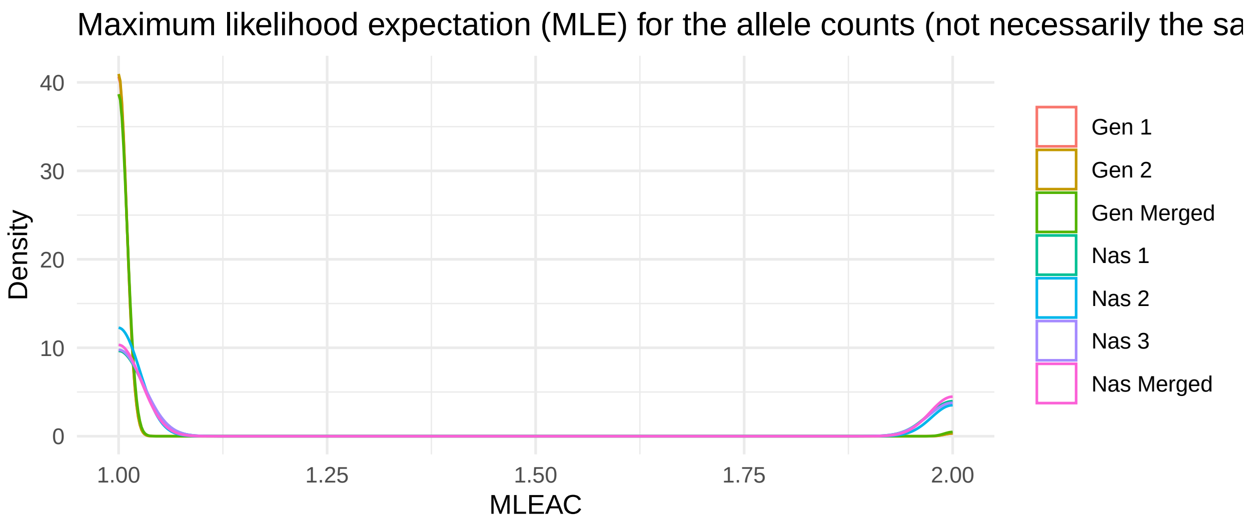
```r
plot_density(full_info_table, "AC", labels, descriptions)
```



```
plot_density(full_info_table, "AF", labels, descriptions)
```

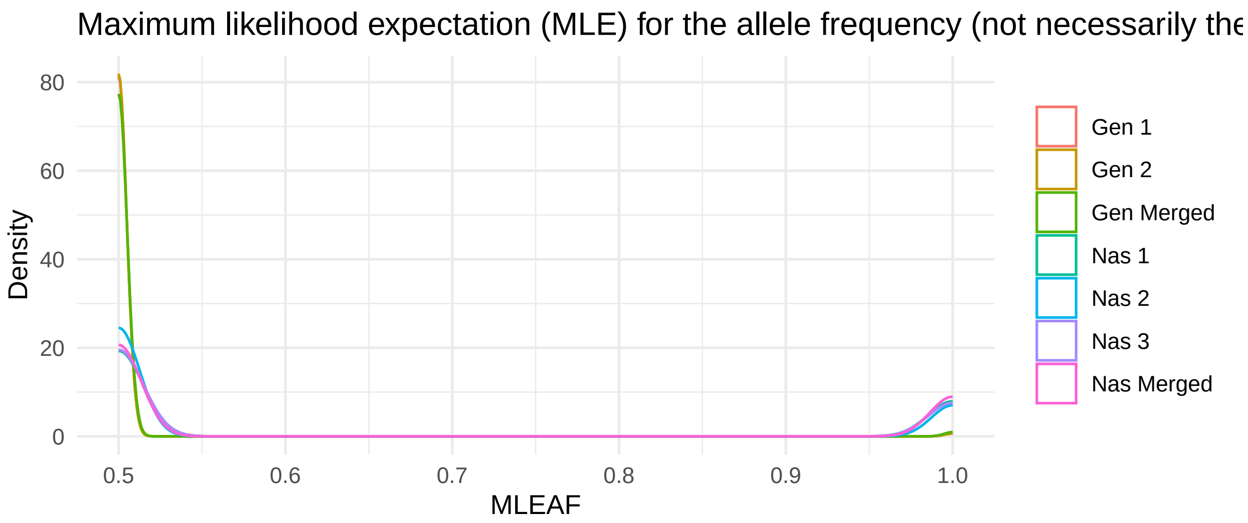


```
plot_density(full_info_table, "MLEAC", labels, descriptions)
```



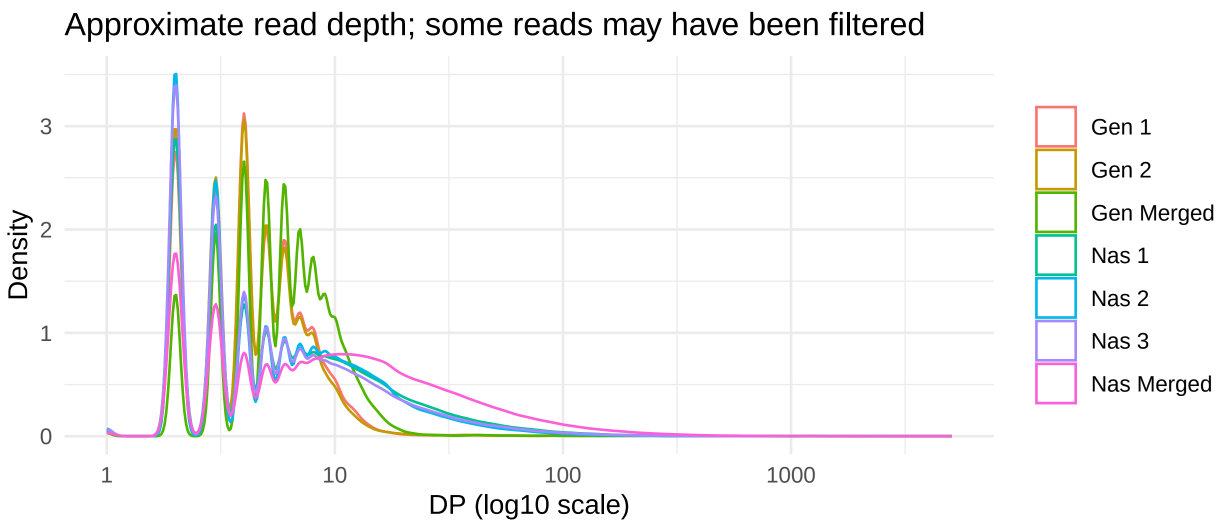
Maximum likelihood expectation (MLE) for the allele counts (not necessarily the sa

```
plot_density(full_info_table, "MLEAF", labels, descriptions)
```

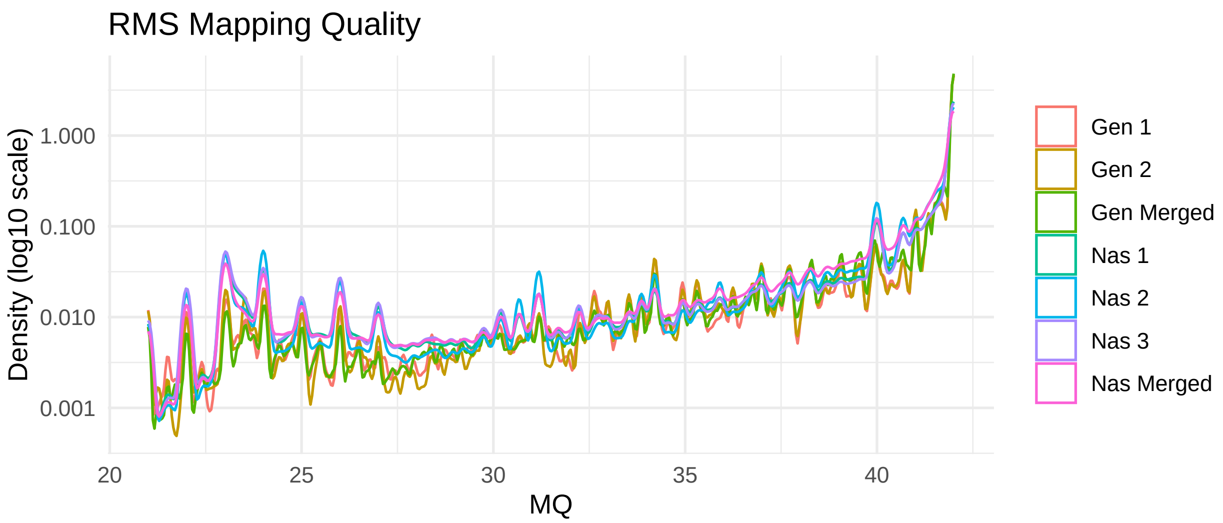


Maximum likelihood expectation (MLE) for the allele frequency (not necessarily the

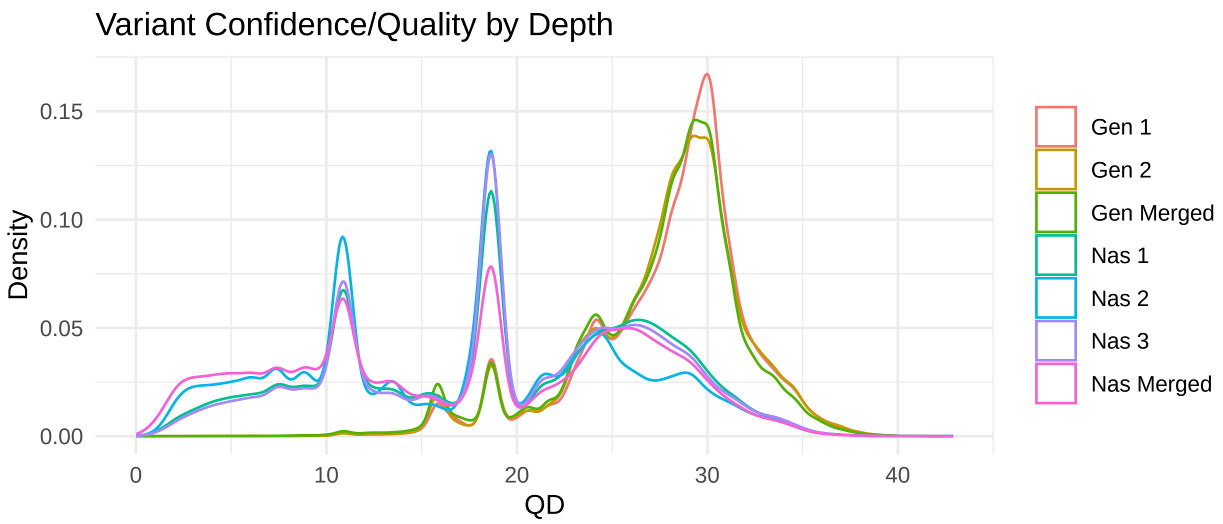
```
plot_density(full_info_table, 'DP', labels, descriptions, x_log10=TRUE)
```



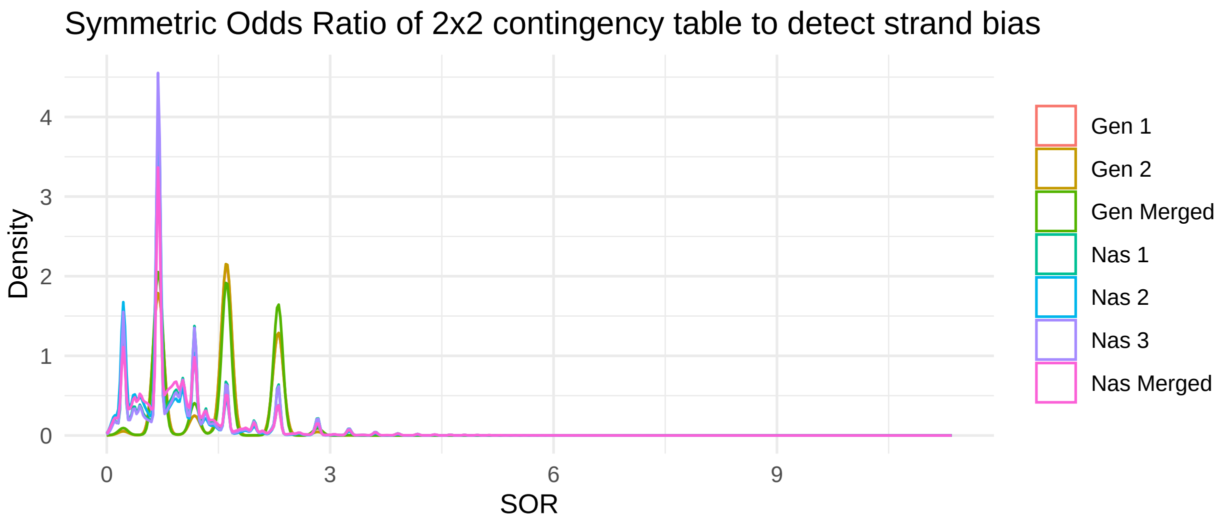
```
plot_density(full_info_table, 'MQ', labels, descriptions, y_log10=TRUE)
```



```
plot_density(full_info_table, 'QD', labels, descriptions)
```



```
plot_density(full_info_table, 'SOR', labels, descriptions)
```



Analysis of intersections

```
gen1_uf <- load_vcf(gen_vcf1_path, hardfilter=FALSE)
gen2_uf <- load_vcf(gen_vcf2_path, hardfilter=FALSE)
gen_merged_uf <- load_vcf(gen_merged_vcf_path, hardfilter=FALSE)
nas1_uf <- load_vcf(nas_vcf1_path, hardfilter=FALSE)
nas2_uf <- load_vcf(nas_vcf2_path, hardfilter=FALSE)
nas3_uf <- load_vcf(nas_vcf3_path, hardfilter=FALSE)
nas_merged_uf <- load_vcf(nas_merged_vcf_path, hardfilter=FALSE)

gen1 <- load_vcf(gen_vcf1_path)
gen2 <- load_vcf(gen_vcf2_path)
gen_merged <- load_vcf(gen_merged_vcf_path, stats=TRUE)
```

```
## [1] "Filtered by FS:  0"
## [1] "Filtered by SOR: 55"
## [1] "Filtered by QUAL:  0"
## [1] "Filtered by MQ:  16514"
## [1] "Filtered by ReadPosRankSum:  0"
## [1] "Filtered by MQRankSum: 0"
```

```
nas1 <- load_vcf(nas_vcf1_path, stats=TRUE)
```

```
## [1] "Filtered by FS:  0"
## [1] "Filtered by SOR: 3808"
## [1] "Filtered by QUAL:  0"
## [1] "Filtered by MQ:  114154"
## [1] "Filtered by ReadPosRankSum:  4"
## [1] "Filtered by MQRankSum: 0"
```

```
nas2 <- load_vcf(nas_vcf2_path, stats=TRUE)
```

```
## [1] "Filtered by FS:  0"
## [1] "Filtered by SOR: 3246"
## [1] "Filtered by QUAL:  0"
## [1] "Filtered by MQ:  163330"
## [1] "Filtered by ReadPosRankSum:  3"
## [1] "Filtered by MQRankSum: 2"
```

```
nas3 <- load_vcf(nas_vcf3_path, stats=TRUE)
```

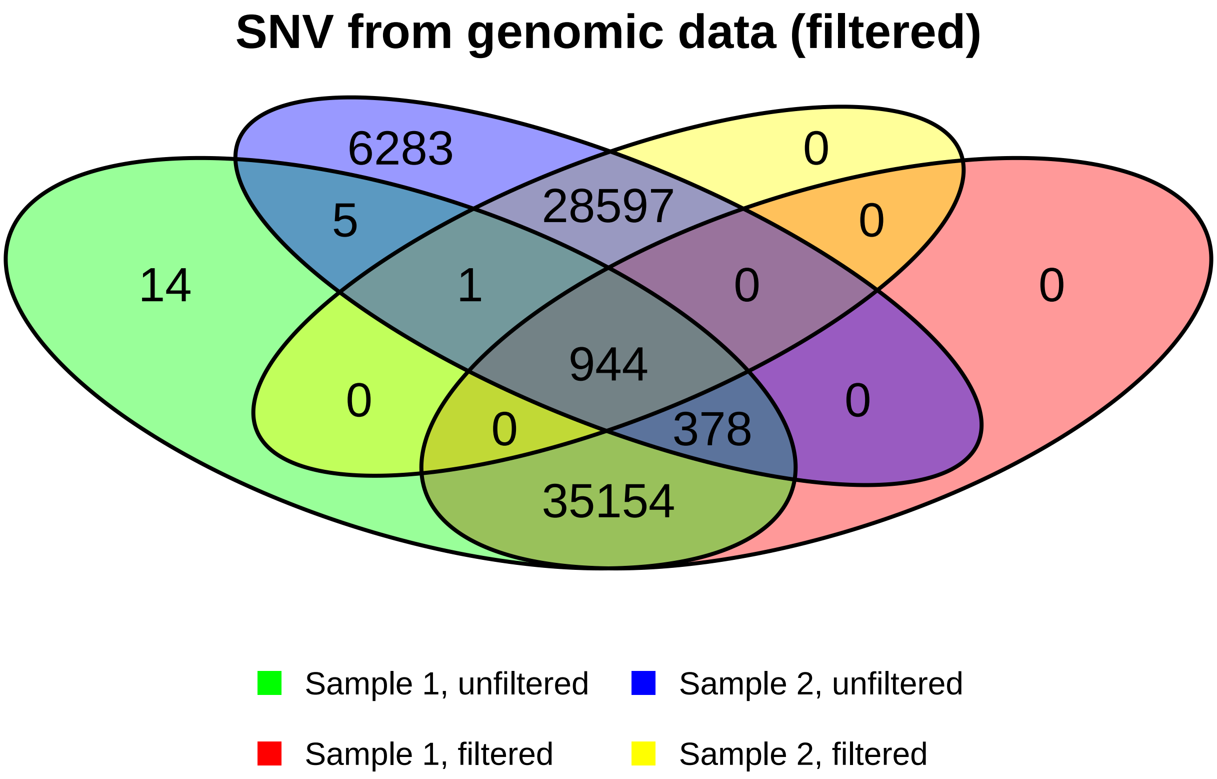
```
## [1] "Filtered by FS:  0"
## [1] "Filtered by SOR: 3260"
## [1] "Filtered by QUAL:  0"
## [1] "Filtered by MQ:  105722"
## [1] "Filtered by ReadPosRankSum:  3"
## [1] "Filtered by MQRankSum: 0"
```

```
nas_merged <- load_vcf(nas_merged_vcf_path, stats=TRUE)
```

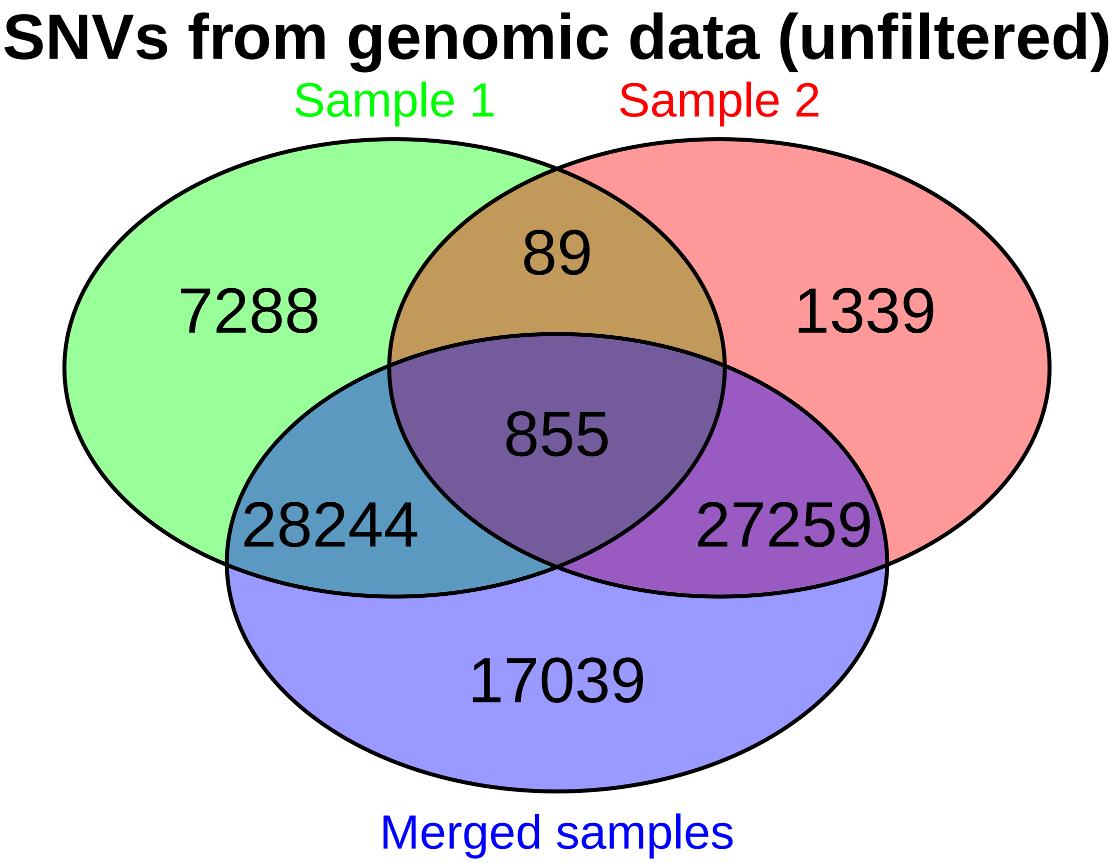
```
## [1] "Filtered by FS:  0"
## [1] "Filtered by SOR: 6291"
## [1] "Filtered by QUAL:  0"
## [1] "Filtered by MQ:  207597"
## [1] "Filtered by ReadPosRankSum:  34"
## [1] "Filtered by MQRankSum: 21"
```

SNVs from Genomic Data

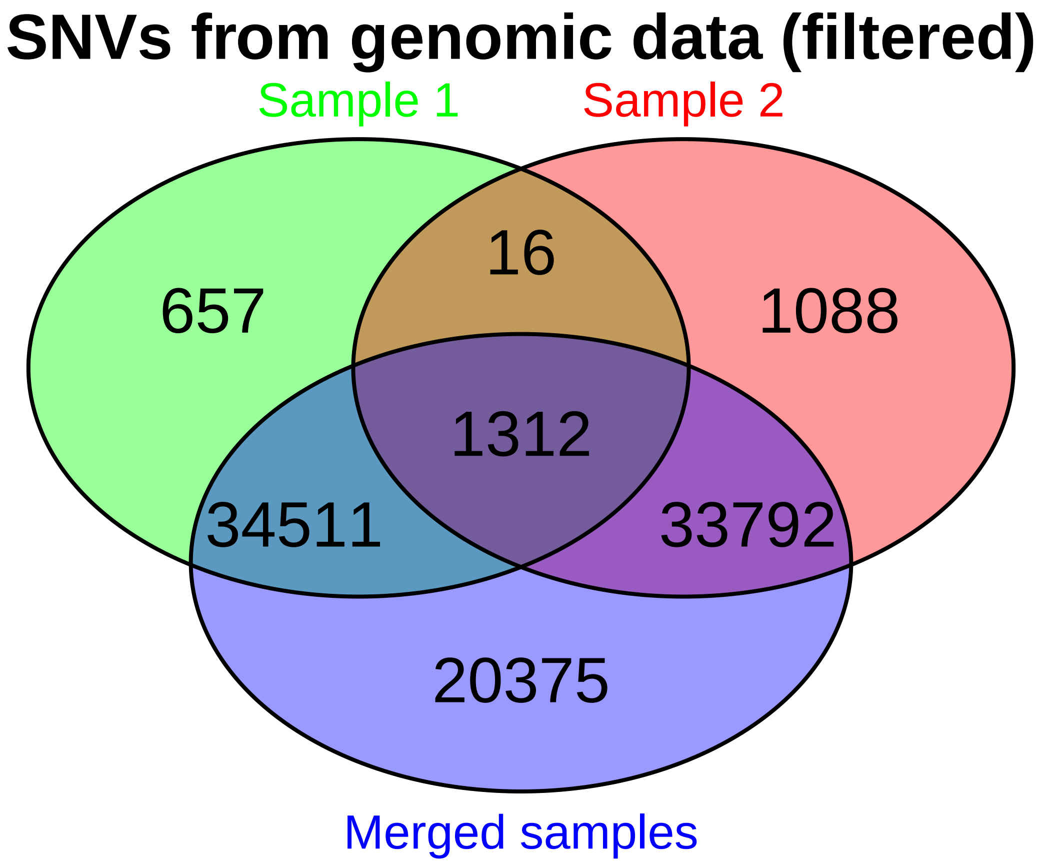
```
plot_venn_4sets(data_list=list(gen1_uf,
                          gen1,
                          gen2_uf,
                          gen2),
            param='All SNPs',
            labels=c("Sample 1, unfiltered",
                     "Sample 1, filtered",
                     "Sample 2, unfiltered",
                     "Sample 2, filtered"),
            title="SNV from genomic data (filtered)")
```



```
plot_venn_3sets(data_list=list(gen1,
                               gen2,
                               gen_merged),
          param='All SNPs',
          labels=c("Sample 1",
                   "Sample 2",
                   "Merged samples"),
          title="SNVs from genomic data (unfiltered)")
```

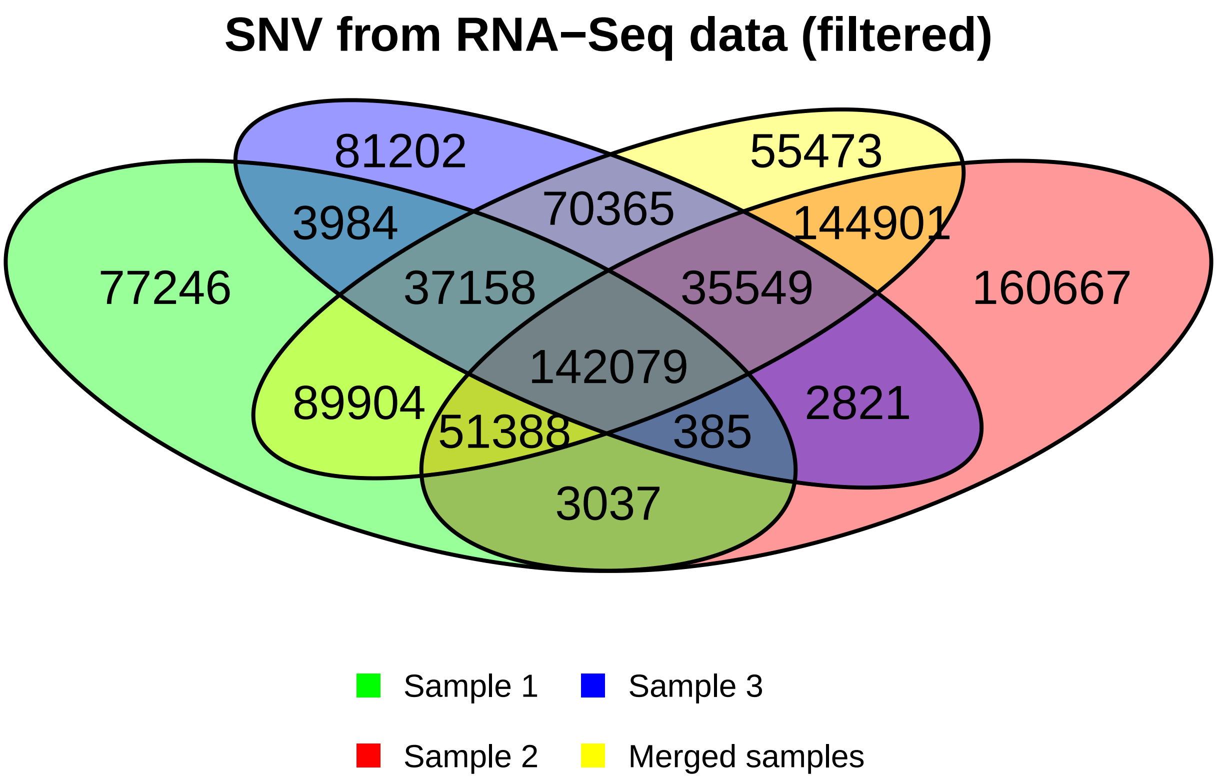


```
plot_venn_3sets(data_list=list(gen1_uf,
                               gen2_uf,
                               gen_merged_uf),
          param='All SNPs',
          labels=c("Sample 1",
                   "Sample 2",
                   "Merged samples"),
          title="SNVs from genomic data (filtered)")
```

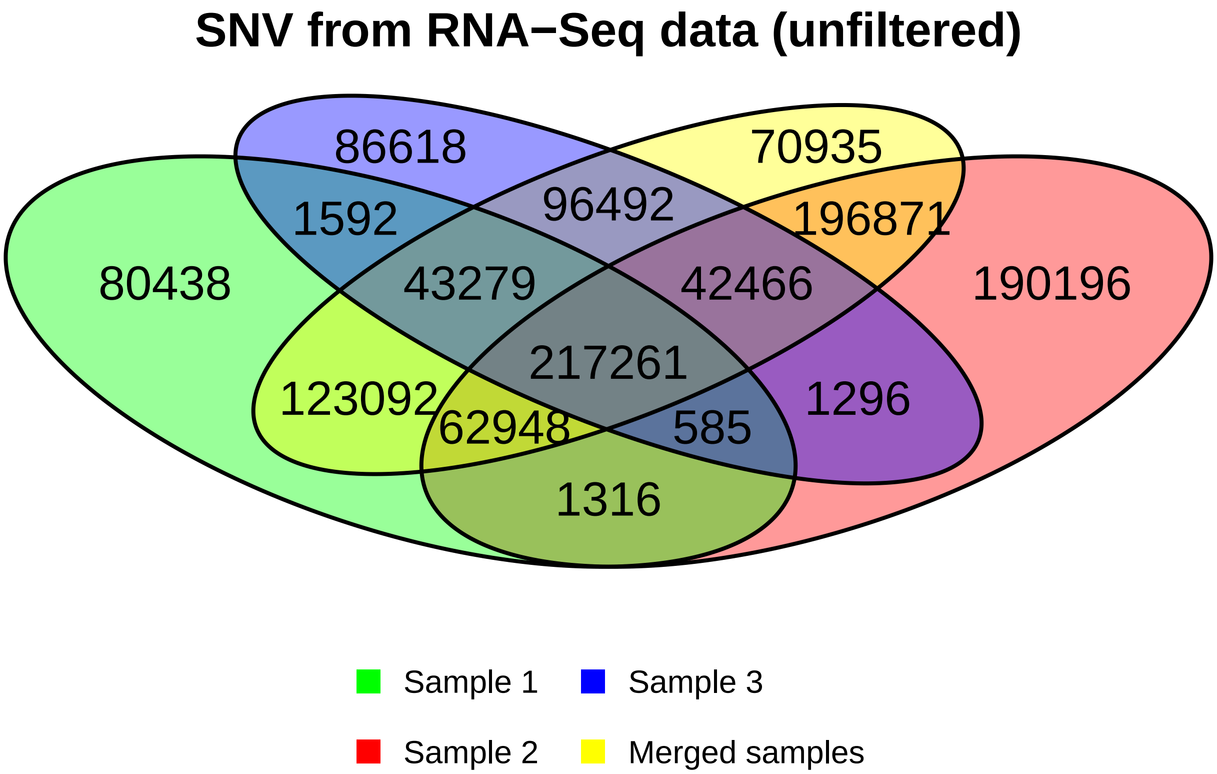


SNVs from RNA-Seq Data

```
plot_venn_4sets(data_list=list(nas1,
                               nas2,
                               nas3,
                               nas_merged),
          param='All SNPs',
          labels=c("Sample 1",
                   "Sample 2",
                   "Sample 3",
                   "Merged samples"),
          title="SNV from RNA-Seq data (filtered)")
```



```
plot_venn_4sets(data_list=list(nas1_uf,
                               nas2_uf,
                               nas3_uf,
                               nas_merged_uf),
          param='All SNPs',
          labels=c("Sample 1",
                   "Sample 2",
                   "Sample 3",
                   "Merged samples"),
          title="SNV from RNA-Seq data (unfiltered)")
```



SNVs from Genomic and RNA-Seq data

Out of 73 thousand SNVs detected in merged genomic data and 627 thousands of SNVs detected in merged nascent RNA data only 12 thousands SNVs are common:

```
length(gen_merged@ranges@NAMES)
```

```
## [1] 73397
```

```
length(nas_merged@ranges@NAMES)
```

```
## [1] 626817
```

```
length(intersect(gen_merged@ranges@NAMES, nas_merged@ranges@NAMES))
```

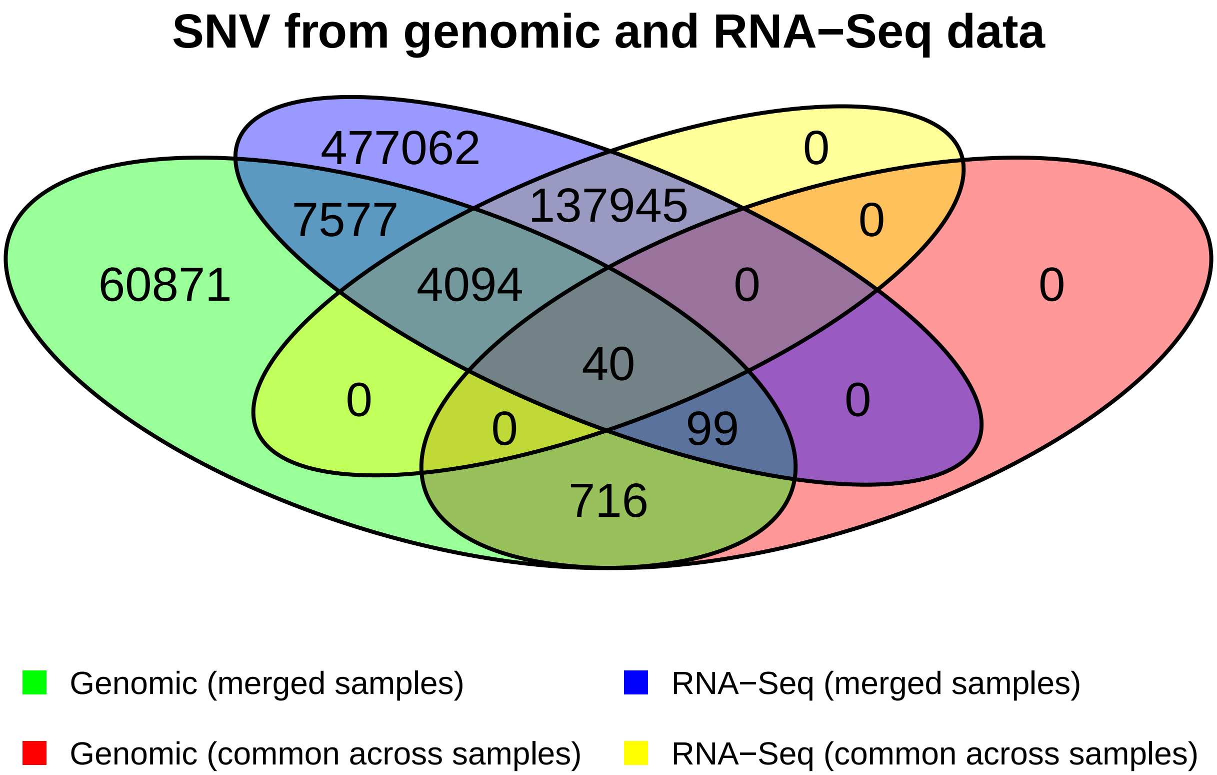
```
## [1] 11810
```

Only 40 SNVs are common between all VCF files used:

```
gen_common <- gen_merged[gen_merged@ranges@NAMES %in%
                           Reduce(intersect, list(gen1@ranges@NAMES,
                                                  gen2@ranges@NAMES,
                                                  gen_merged@ranges@NAMES))]

nas_common <- nas_merged[nas_merged@ranges@NAMES %in%
                           Reduce(intersect, list(nas1@ranges@NAMES,
                                                  nas2@ranges@NAMES,
                                                  nas3@ranges@NAMES,
                                                  nas_merged@ranges@NAMES))]
```

```
plot_venn_4sets(data_list=list(gen_merged,
                               gen_common,
                               nas_merged,
                               nas_common),
          param='All SNPs',
          labels=c("Genomic (merged samples)",
                   "Genomic (common across samples)",
                   "RNA-Seq (merged samples)",
                   "RNA-Seq (common across samples)"),
          title="SNV from genomic and RNA-Seq data")
```

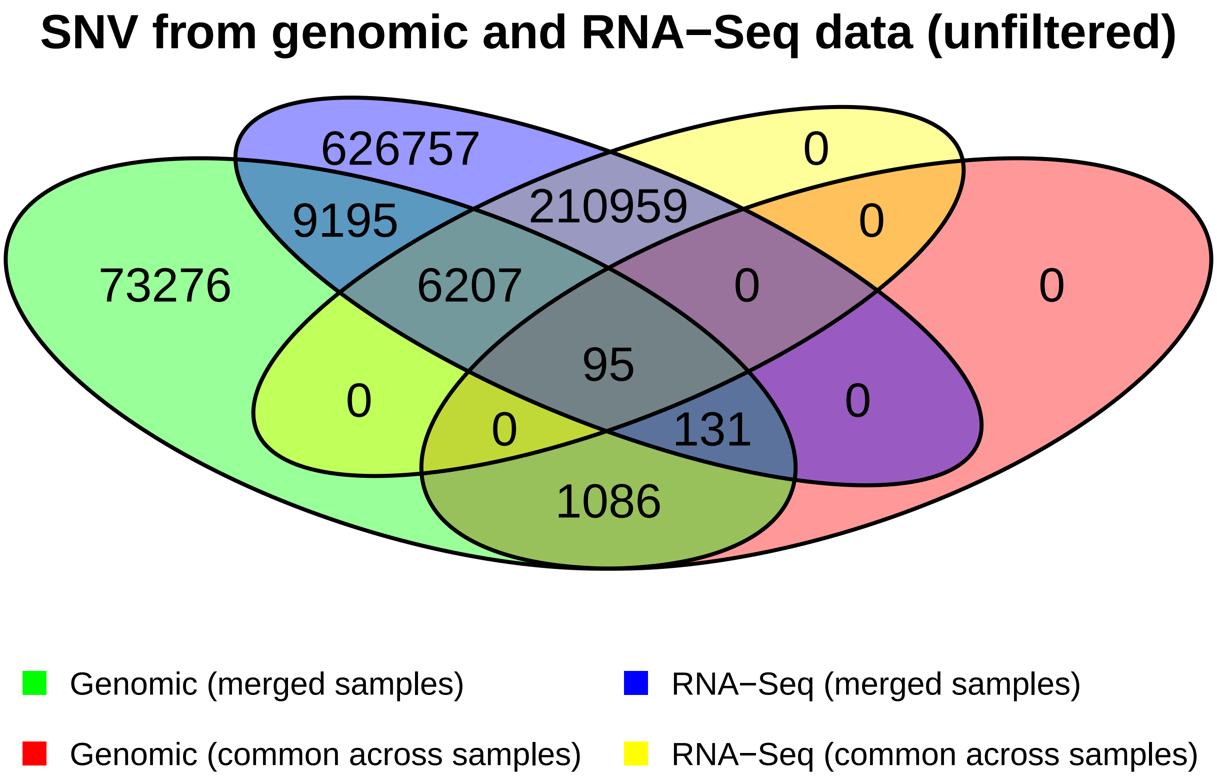


In case of unfiltered data, this intersection was only 95 SNVs:

```
gen_common_uf <- gen_merged_uf[gen_merged_uf@ranges@NAMES %in%
                           Reduce(intersect, list(gen1_uf@ranges@NAMES,
                                                   gen2_uf@ranges@NAMES,
                                                   gen_merged_uf@ranges@NAMES))]
```

```
nas_common_uf <- nas_merged_uf[nas_merged_uf@ranges@NAMES %in%
                           Reduce(intersect, list(nas1_uf@ranges@NAMES,
                                                  nas2_uf@ranges@NAMES,
                                                  nas3_uf@ranges@NAMES,
                                                  nas_merged_uf@ranges@NAMES))]

plot_venn_4sets(data_list=list(gen_merged_uf,
                               gen_common_uf,
                               nas_merged_uf,
                               nas_common_uf),
          param='All SNPs',
         labels=c("Genomic (merged samples)",
                   "Genomic (common across samples)",
                   "RNA-Seq (merged samples)",
                   "RNA-Seq (common across samples)"),
          title="SNV from genomic and RNA-Seq data (unfiltered)")
```



Therefore, SNVs which are common in all nascent RNA VCF files will be used further.

```
common_snps <- Reduce(intersect, list(nas1@ranges@NAMES,
                                      nas2@ranges@NAMES,
                                      nas3@ranges@NAMES,
                                      nas_merged@ranges@NAMES))
```

Comparison of unique and intersecting SNVs

Define subsets of SNVs which are unique for all the files (e.g. SNVs which are present in gen_vcf1, but absent in gen_vcf2, gen_merged, nas_vcf1, nas_vcf2, nas_vcf3, nas_merged) and visualise stats.

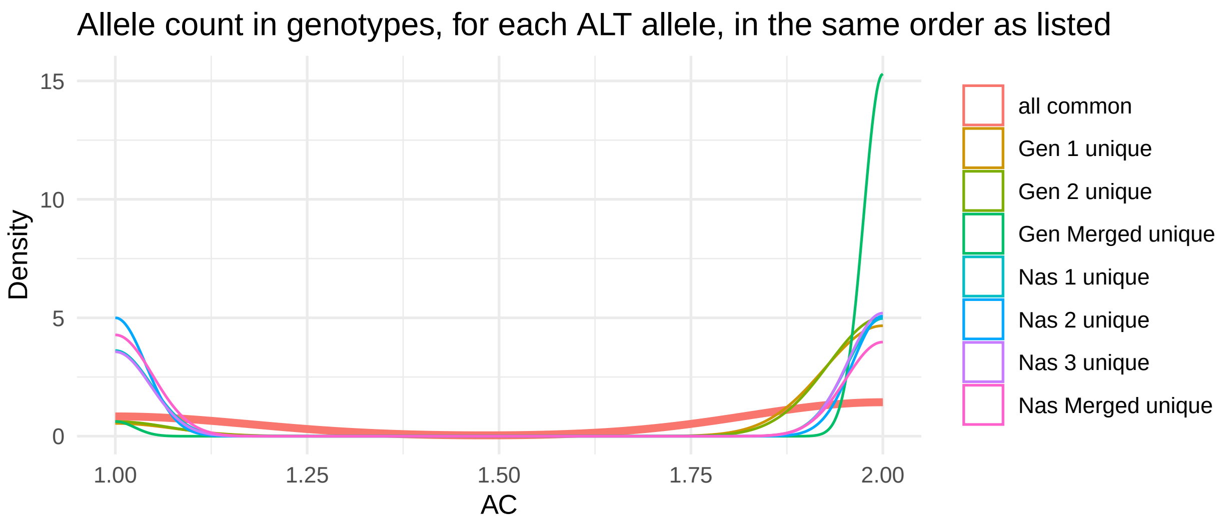
```r
# Extract INFO tables and SNV positions (i.e. SNV identifiers)
info_tables <- lapply(filepaths, extract_info_and_positions)
```

```
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
## Warning in .bcfHeaderAsSimpleList(header): duplicate keys in header will be
## forced to unique rownames
```

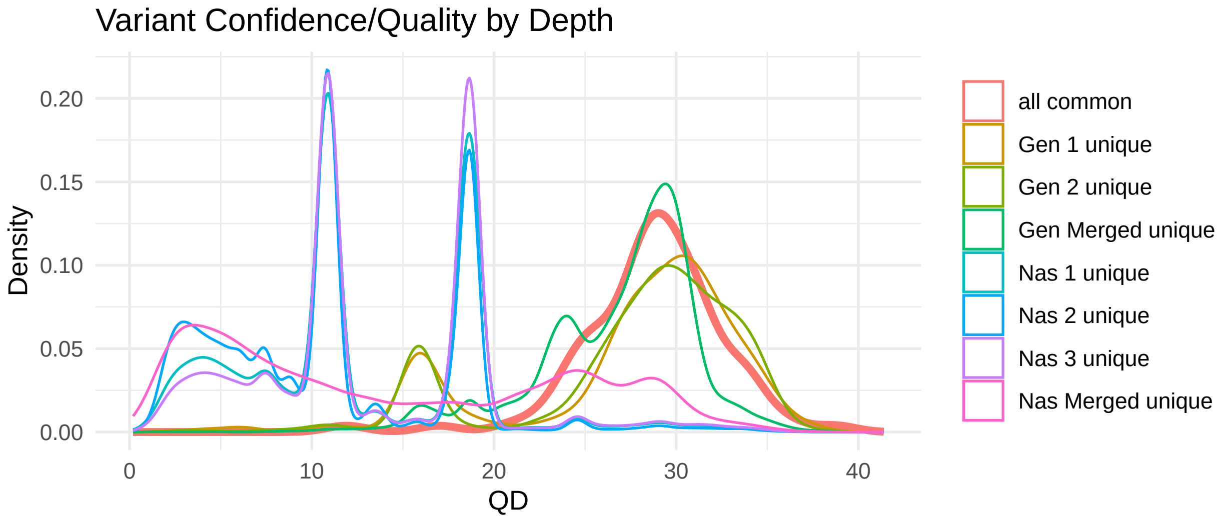
```r
# Identify unique and common SNV sets based on positions
snv_set_positions <- identify_snv_sets_positions(lapply(info_tables, function(x) x$positions),
                                                 labels)
```

```r
param_name <- "AC"
snv_set_with_metric <- extract_metric_from_info_tables(info_tables, param_name, snv_set_positions)
plot_ac <- plot_snv_density(snv_set_with_metric, param_name, descriptions_df)
print(plot_ac)
```

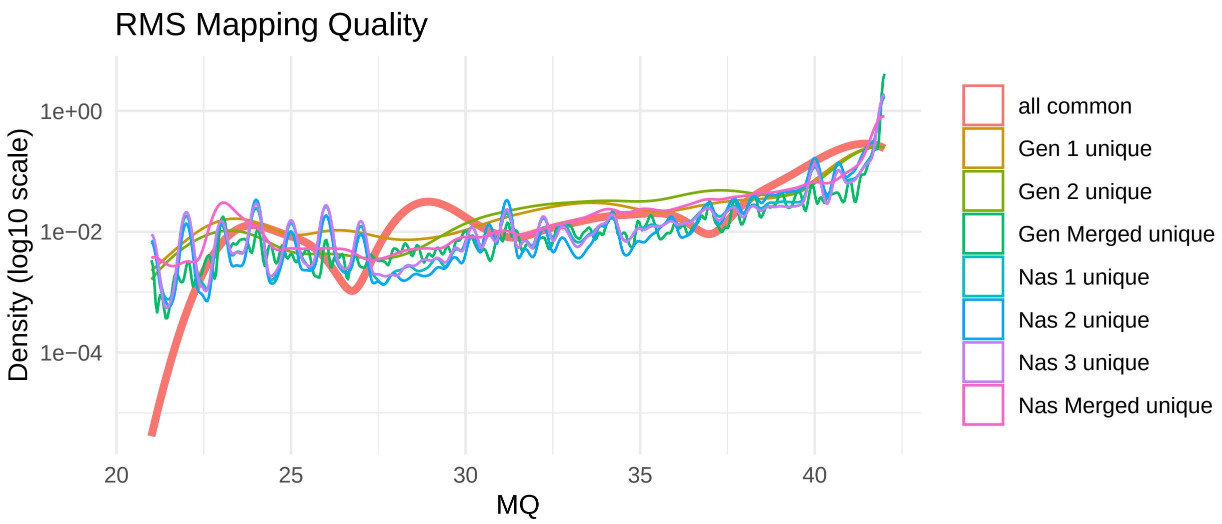
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



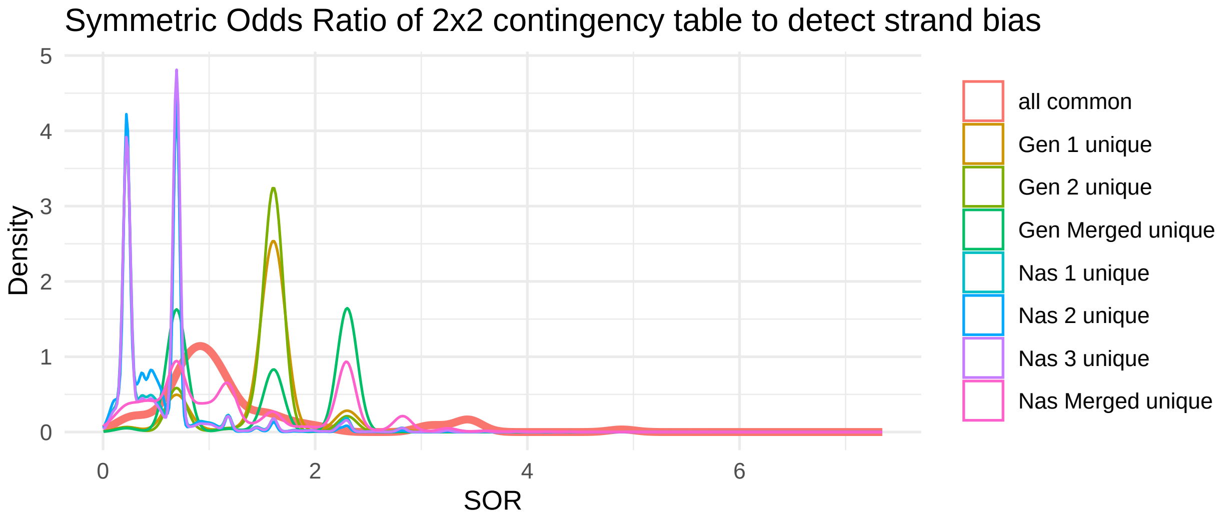
```
param_name <- "QD"
snv_set_with_metric <- extract_metric_from_info_tables(info_tables, param_name, snv_set_positions)
plot_qd <- plot_snv_density(snv_set_with_metric, param_name, descriptions_df)
print(plot_qd)
```



```
param_name <- "MQ"
snv_set_with_metric <- extract_metric_from_info_tables(info_tables, param_name, snv_set_positions)
plot_mq <- plot_snv_density(snv_set_with_metric, param_name, descriptions_df, x_log10 = FALSE, y_log10 =
print(plot_mq)
```



```
param_name <- "SOR"
snv_set_with_metric <- extract_metric_from_info_tables(info_tables, param_name, snv_set_positions)
plot_sor <- plot_snv_density(snv_set_with_metric, param_name, descriptions_df, x_log10 = FALSE, y_log10
print(plot_sor)
```


Symmetric Odds Ratio of 2x2 contingency table to detect strand bias
Density
SOR
all common
Gen 1 unique
Gen 2 unique
Gen Merged unique
Nas 1 unique
Nas 2 unique
Nas 3 unique
Nas Merged unique