Eplet Comparison Workflow

Patricia Hernandez Nick Borcherding

2025-05-30

Table of contents

1 Loading Library

This will load the libraries necessary to run the analysis. If the libraries/packages are not installed, this code will install them before loading them.

```r
required_libraries <- c(
  "dplyr", "data.table", "tidyr", "janitor", "stringr", "venn",
  "lubridate", "readxl", "openxlsx", "tidyverse", "hrbrthemes",
  "proustr", "VennDiagram", "ComplexUpset", "ggVennDiagram"
)

# Function to check, install if needed, and load libraries
load_libraries <- function(libraries) {
  for (lib in libraries) {
    if (!requireNamespace(lib, quietly = TRUE)) {
      install.packages(lib, dependencies = TRUE)
    }
    library(lib, character.only = TRUE)
  }
}

# Execute the function
load_libraries(required_libraries)
```

2 Loading Functions

These are custom functions written for the analysis to assist in loading and plotting the data. More information can be found in the github repository under the R folder.

```r
source("./R/load_data.R")
source("./R/utils.R")
source("./R/calculate_sum.R")
source("./R/generate_venn_diagram.R")
source("./R/generate_upset_plot.R")
source("./R/summarise_eplet.R")
source("./R/plot_eplet_overlap.R")
source("./R/plot_sab_overlap.R")
```

3 Data Formats

The most important step before running the analysis is to ensure that the data is consistently formatted. Much of the functions are specific to the column names and structure of the individual cells within the excel file or the document.

3.1 SAB Assay

```r
SAB <- read_xlsx("data/SAB_Class_I_Lot_14_20221212.xlsx", sheet = 1)
head(SAB)
```

```
# A tibble: 6 x 11
  Column1 `Bead ID` `Antigen ID` `Molecular Typing` `Serological Typing`
    <dbl>     <dbl> <chr>        <chr>              <chr>
1       1         1 NC           NA                 N/A
2       2         2 PC           NA                 N/A
3       3         3 rA0101       A*01:01            A1
4       4         4 rA0201       A*02:01            A2
5       5         5 rA0203       A*02:03            A2
6       6         6 rA0206       A*02:06            A2
# i 6 more variables: `*W6/32` <dbl>, Results <lgl>, Column8 <chr>,
#   Column9 <chr>, Column10 <chr>, Column11 <chr>
```

3.2 PRA Assay

```r
PRA <- read_xlsx("data/PRA_Class_I_Lot_020_20221212.xlsx", sheet = 1)
head(PRA)
```

```
# A tibble: 6 x 11
  Order `Bead ID` `Antigen ID` `A Left` `A Right`    `B Left` `B Right` `C Left`
  <dbl> <chr>     <chr>        <chr>    <chr>        <chr>    <chr>     <chr>
1     1 1         NC           N/A      N/A          N/A      N/A       N/A
2     2 2         PC           N/A      N/A          N/A      N/A       N/A
3     3 3         C4966        A*02:01  A*02:07      B*46:01  X         C*01:02
4     4 4         E19109       A*01:01  A*23:01      B*49:01  B*55:01   C*03:03
5     5 5         G0142        A*11:02  A*24:02      B*27:06  B*40:01   C*03:04
6     6 6         E5482        A*11:01  A*24:02/50/~ B*54:01  B*59:01   C*01:02
# i 3 more variables: `C Right` <chr>, `Serological Typing` <chr>,
#   Results <lgl>
```

3.3 Mix Assay

```r
Mix <- read_xlsx("data/LSM12NC23_024_01.xlsx", sheet = 1)
head(Mix)
```

```
# A tibble: 6 x 8
   Bead `Pos Ctr` Abbreviated Specificit~1 ...4  `Recognition Site` ...6  Cutoff
  <dbl> <lgl>     <chr>                    <lgl> <lgl>              <lgl> <lgl>
1     1 NA        -,-,-,-,-,-,-,-          NA    NA                 NA    NA
2     2 NA        -,-,-,-,-,-,-,-          NA    NA                 NA    NA
```

```
3     6 NA        A1,A80,B18,B50,Bw6,-,Cw~ NA    NA                NA    NA
4     6 NA        A1,A29,B8,B45,Bw6,-,Cw6~ NA    NA                NA    NA
5     6 NA        A1,A69,B8,B55,Bw6,-,Cw1~ NA    NA                NA    NA
6     9 NA        A11,A24,B35,B62,Bw6,-,C~ NA    NA                NA    NA
# i abbreviated name: 1: `Abbreviated Specificity`
# i 1 more variable: Specificity <chr>
```

3.4 ExPlex Assay

```
ExPlex <- read_xlsx("data/ExPlex_classI_II.xlsx", sheet = 1)
head(ExPlex)
```

```
# A tibble: 6 x 1
  Molecular.Typing
  <chr>
1 A*01:02
2 A*02:10
3 A*02:05
4 A*02:07
5 A*02:18
6 A*26:02
```

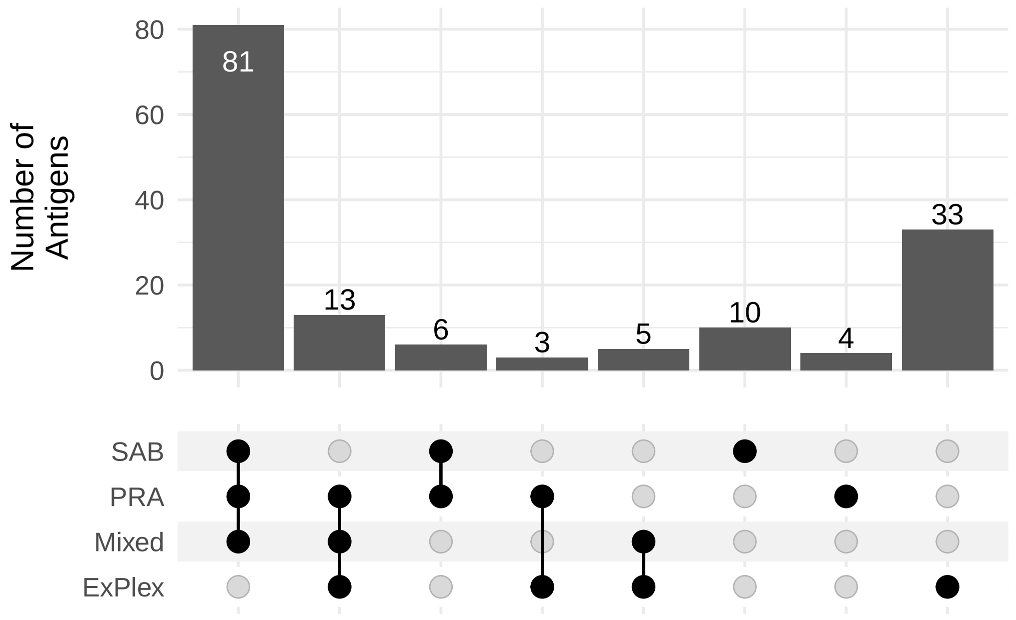
4 Figure 1: Allele Counting

The following code will load all the assay information and automatically format them with the `load_XX_data()` functions. From there, we can count the alleles.

4.1 Class I Alleles

```
sab <- process_sab_data(file = "./data/SAB_Class_I_Lot_14_20221212.xlsx", c(1,2))
pra <- process_pra_data(file = "./data/PRA_Class_I_Lot_020_20221212.xlsx", 1)
mix <- process_mix_data(file = "./data/LSM12NC23_024_01.xlsx", 1)
explex <- process_explex_data(file = "./data/ExPlex_classI_II.xlsx", 1)

#Remove non-type-I from Mix
mix <- mix[grepl("A_|B_|C_", mix$values),]

b <- list(
  SAB = unique(sab$Molecular.Typing),
  PRA = unique(pra$values),
  Mixed = unique(mix$values),
  ExPlex = unique(explex$Molecular.Typing)
```

```r
)

# Create a binary membership matrix
all_elements <- unique(unlist(b))
membership_matrix <- data.frame(
  element = all_elements,
  SAB = all_elements %in% b$SAB,
  PRA = all_elements %in% b$PRA,
  Mixed = all_elements %in% b$Mix,
  ExPlex = all_elements %in% b$ExPlex
)

# Convert logical values to binary
membership_matrix <- membership_matrix %>%
  select(element, ExPlex, Mixed, PRA, SAB)

# Generate the UpSet plot
upset_data <- membership_matrix %>% select(-element)

upset_data <- upset_data[, c("ExPlex", "Mixed", "PRA", "SAB")]

# Then specify the same order in the `intersect` argument:
upset(
  upset_data,
  intersect = colnames(upset_data), # Now in alphabetical order
  sort_intersections_by='degree',
  base_annotations = list('Number of\nAntigens' = intersection_size()),
  themes = upset_default_themes(text = element_text(size = 12)),
  set_sizes = FALSE,
  sort_sets = FALSE
) +
  theme(axis.title.x = element_blank())
```

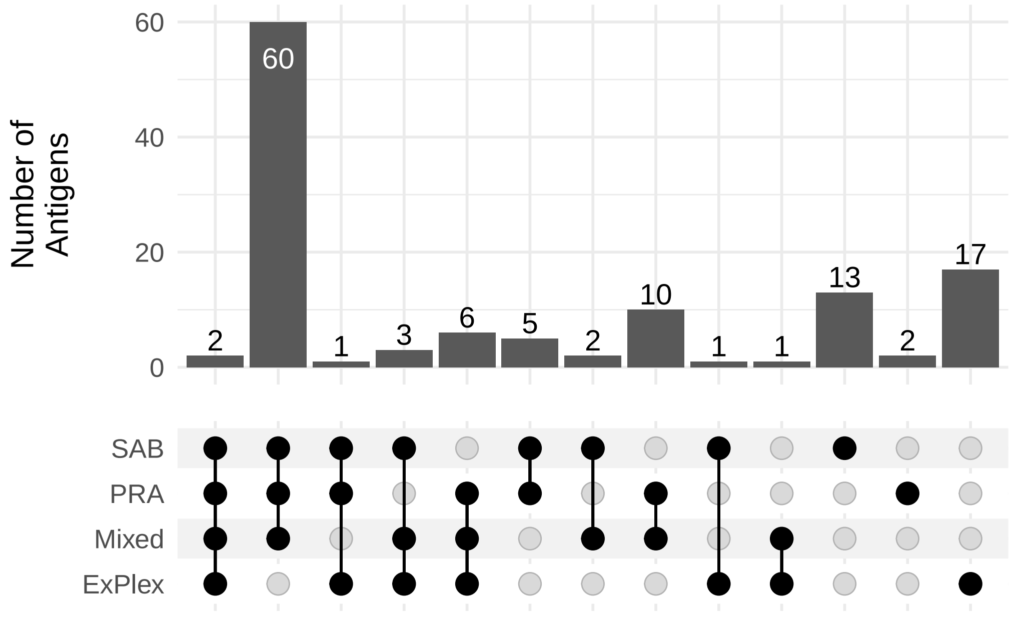


```r
ggsave('outputs/viz/Figure1_ClassI.pdf', width = 8, height = 6)

# Summarizing results for supplemental table
summary_table <- membership_matrix %>%
  pivot_longer(cols = -element, names_to = "Assay", values_to = "Present") %>%
  filter(Present) %>%
  group_by(element) %>%
  summarize(Assay_Combination = paste(sort(Assay), collapse = "_")) %>%
  group_by(Assay_Combination) %>%
  summarize(
    Allele_Count = n(),
    Alleles = paste(sort(element), collapse = ", ")
  ) %>%
  arrange(desc(Allele_Count))
write.csv(summary_table, "outputs/files/SupplementalTable1_ClassI_Allele_Overlap_Table.csv", r
```

4.2 Class II Alleles

```r
sab <- process_sab_data(file = "./data/SAB_ClassII_Lot_15_20221212.xlsx", c(1:2))[-c(1:2),]
pra <- process_pra_data(file = "./data/PRA_Class_II_Lot_019_20221212.xlsx", 1)
mix <- process_mix_data(file = "./data/LSM12NC23_024_01.xlsx", 1)
explex <- process_explex_data(file = "./data/ExPlex_classI_II.xlsx", 2)

#Remove type-I from Mix
mix <- mix[!grepl("A_|B_|C_", mix$values),]
```

```r
b <- list(
  SAB = unique(sab$Molecular.Typing),
  PRA = unique(pra$values),
  Mixed = unique(mix$values),
  ExPlex = unique(explex$Molecular.Typing)
)

# Create a binary membership matrix
all_elements <- unique(unlist(b))
membership_matrix <- data.frame(
  element = all_elements,
  SAB = all_elements %in% b$SAB,
  PRA = all_elements %in% b$PRA,
  Mixed = all_elements %in% b$Mix,
  ExPlex = all_elements %in% b$ExPlex
)

# Convert logical values to binary
membership_matrix <- membership_matrix %>%
  select(element, ExPlex, Mixed, PRA, SAB)

# Generate the UpSet plot
upset_data <- membership_matrix %>% select(-element)

upset_data <- upset_data[, c("ExPlex", "Mixed", "PRA", "SAB")]

# Then specify the same order in the `intersect` argument:
upset(
  upset_data,
  intersect = colnames(upset_data),# Now in alphabetical order
  sort_intersections_by='degree',
  base_annotations = list('Number of\nAntigens' = intersection_size()),
  themes = upset_default_themes(text = element_text(size = 12)),
  set_sizes = FALSE,
  sort_sets = FALSE
) +
  theme(axis.title.x = element_blank())
```



```r
ggsave('outputs/viz/Figure1_ClassII.pdf', width = 8, height = 6)

# Summarizing results for supplemental table
summary_table <- membership_matrix %>%
  pivot_longer(cols = -element, names_to = "Assay", values_to = "Present") %>%
  filter(Present) %>%
  group_by(element) %>%
  summarize(Assay_Combination = paste(sort(Assay), collapse = "_")) %>%
  group_by(Assay_Combination) %>%
  summarize(
    Allele_Count = n(),
    Alleles = paste(sort(element), collapse = ", ")
  ) %>%
  arrange(desc(Allele_Count))
write.csv(summary_table, "outputs/files/SupplementalTable2_ClassII_Allele_Overlap_Table.csv", 
```

5 Figure 2: Eplet Counting

Here we are going to calculate the respective eplet load within the individual assays. For each assay, we use the `calculate_sum()` function to add up the unique eplets represented by the alleles within the assay.

5.1 Class I Eplets

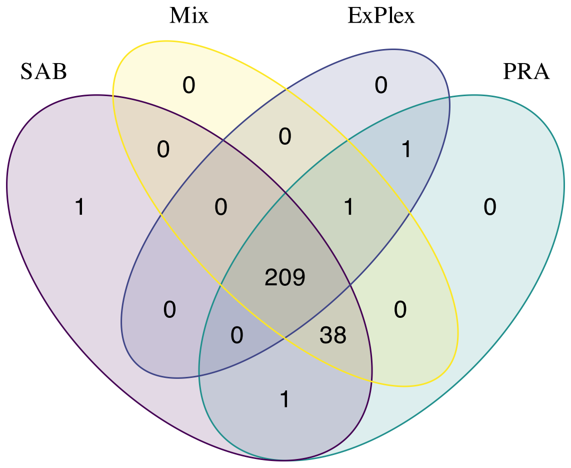
```
# Load and preprocess data
df <- read.csv("./data/EpletRegistry_ClassI.csv")

sab <- process_sab_data(file = "./data/SAB_Class_I_Lot_14_20221212.xlsx", c(1,2))
pra <- process_pra_data(file = "./data/PRA_Class_I_Lot_020_20221212.xlsx", 1)
mix <- process_mix_data(file = "./data/LSM12NC23_024_01.xlsx", 1)
explex <- process_explex_data(file = "./data/ExPlex_classI_II.xlsx", 1)

# Add allele columns and calculate sums
df <- calculate_sum(df, sab$Molecular.Typing, "SAB_unique")
df <- calculate_sum(df, pra$values, "PRA_unique")
df <- calculate_sum(df, mix$values, "Mix_unique")
df <- calculate_sum(df, explex$Molecular.Typing, "EXPLEX_unique")

# Generate Venn diagrams
generate_venn_diagram(df, "outputs/viz/Figure2_classI.svg",
                      pattern = "_unique")
```

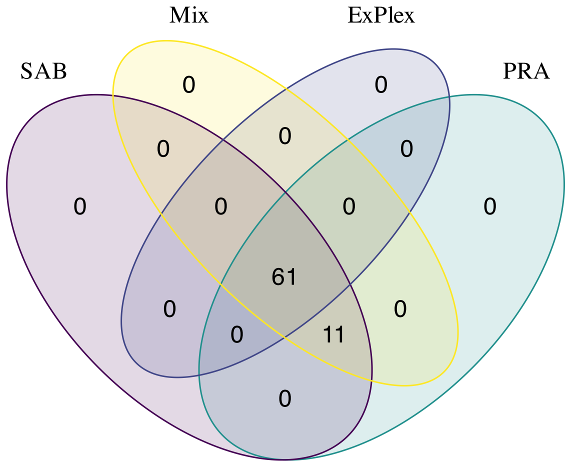
```
[1] 1
```

```
# Filter antibody-verified data and generate another Venn diagram
dfa <- df %>% filter(Antibody.Reactivity == "Confirmed")
generate_venn_diagram(dfa, "outputs/viz/Figure2_classI_AB.svg",
                      pattern = "_unique")
```

```
[1] 1
```

```
# Generating supplemental tables
summary_table <- summarise_eplet(df)
write.csv(summary_table, "outputs/files/SupplementalTable3_ClassI_Eplet_Overlap_Table.csv", ro
summary_table <- summarise_eplet(dfa)
write.csv(summary_table, "outputs/files/SupplementalTable5_ClassI_Eplet_Overlap_Table.csv", ro
```





5.2 Class II Eplets

```
df <- read.csv("./data/EpletRegistry_ClassII.csv")
sab <- process_sab_data(file = "./data/SAB_ClassII_Lot_15_20221212.xlsx", c(1:2))[-c(1:2),]
pra <- process_pra_data(file = "./data/PRA_Class_II_Lot_019_20221212.xlsx", 1)
mix <- process_mix_data(file = "./data/LSM12NC23_024_01.xlsx", 1)
explex <- process_explex_data(file = "./data/ExPlex_classI_II.xlsx", 2)

# Add allele columns and calculate sums
df <- calculate_sum(df, sab$Molecular.Typing, "SAB_unique")
df <- calculate_sum(df, pra$values, "PRA_unique")
df <- calculate_sum(df, mix$values, "Mix_unique")
```

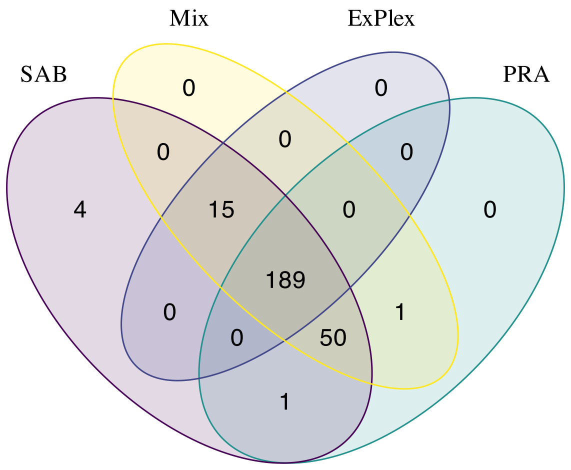
```r
df <- calculate_sum(df, explex$Molecular.Typing, "EXPLEX_unique")

# Generate Venn diagrams
generate_venn_diagram(df, "outputs/viz/Figure2_classII.svg", pattern = "_unique")
```

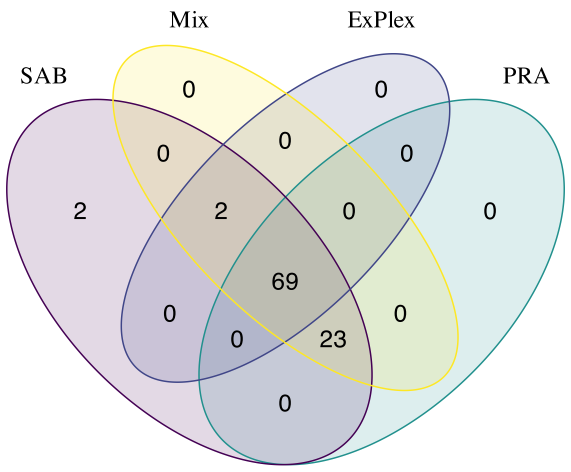
```
[1] 1
```

```r
# Filter antibody-verified data and generate another Venn diagram
dfa <- df %>% filter(Antibody.Reactivity == "Confirmed")
generate_venn_diagram(dfa, "outputs/viz/Figure2_classII_AB.svg", pattern = "_unique")
```

```
[1] 1
```

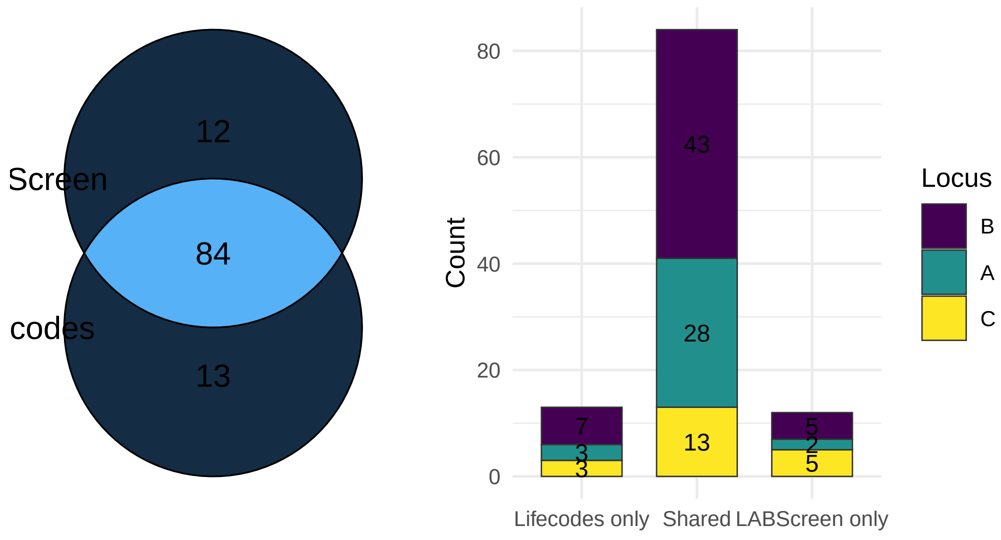
```r
# Generating supplemental tables
summary_table <- summarise_eplet(df)
write.csv(summary_table, "outputs/files/SupplementalTable4_ClassII_Eplet_Overlap_Table.csv", ro
summary_table <- summarise_eplet(dfa)
write.csv(summary_table, "outputs/files/SupplementalTable6_ClassII_Eplet_Overlap_Table.csv", ro
```



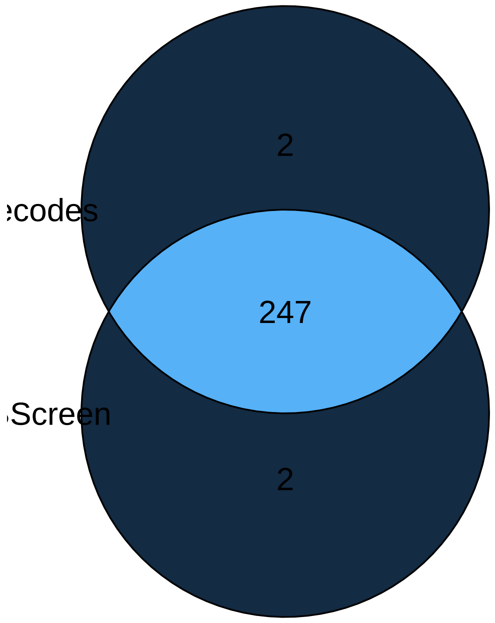


6 Figure 3: SAB Comparisons

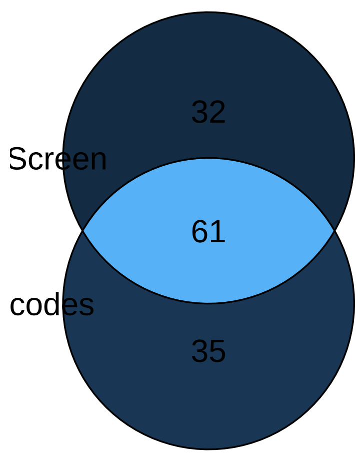
```r
# Class I Analysis----------------------------------------
df <- read.csv("./data/EpletRegistry_ClassI.csv")
OL_ClassI_sab <- process_sab_data(file = "./data/SAB_Class_I_Lot_14_20221212.xlsx", c(1,2))
LC_ClassI_sab <- read.csv("./data/Life_Code_SA1.csv")
LC_ClassI_sab$Molecular.Typing <- gsub("[*]", "_", LC_ClassI_sab$Molecular.Typing)
OL_ClassI_sab$`Molecular Typing` <- gsub("[*]", "_", OL_ClassI_sab$`Molecular Typing`)


# Plot Overlap and Loci
plot_sab_overlap(OL_ClassI_sab$`Molecular Typing`,
                 LC_ClassI_sab$Molecular.Typing,
                 prefix_len = 1)
```

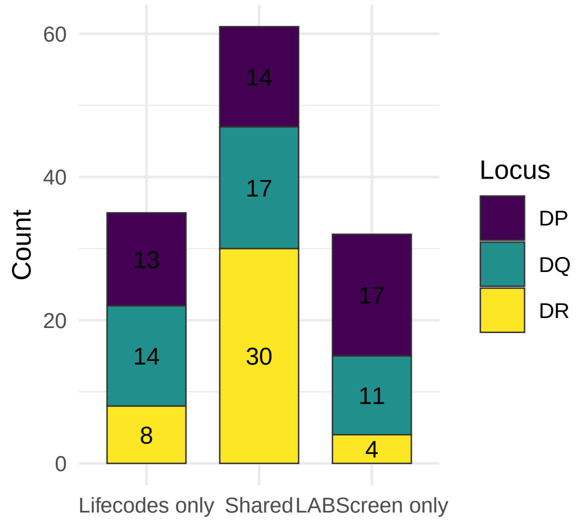


```r
ggsave('outputs/viz/Figure3_classI.pdf',
       width = 8,
       height = 6)

# Performing Eplet Analysis
df <- calculate_sum(df, OL_ClassI_sab$Molecular.Typing, "OL_unique")
df <- calculate_sum(df, LC_ClassI_sab$Molecular.Typing, "LC_unique")

plot_eplet_overlap(df$OL_unique,
                   df$LC_unique,
                   n1         = "LABScreen",
                   n2         = "Lifecodes",
                   thr        = 1)
```

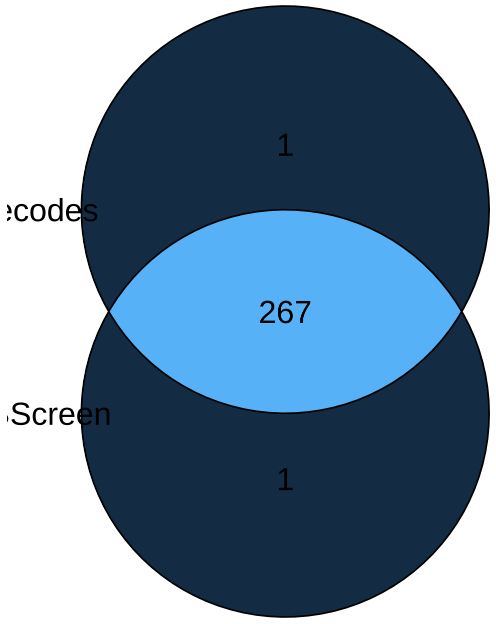


```
ggsave('outputs/viz/Figure3_classI_eplets.pdf', width = 4, height = 4)

# Class II Analysis---------------------------------------
OL_ClassII_sab <- process_sab_data(file = "./data/SAB_ClassII_Lot_15_20221212.xlsx", c(1:2))[-
LC_ClassII_sab <- read.csv("./data/Life_Code_SA2.csv")
LC_ClassII_sab$Molecular.Typing <- gsub("[*]", "_", LC_ClassII_sab$Molecular.Typing)
OL_ClassII_sab$`Molecular Typing` <- gsub("[*]", "_", OL_ClassII_sab$`Molecular Typing`)


# Plot Overlap and Loci
plot_sab_overlap(LC_ClassII_sab$Molecular.Typing,
                 OL_ClassII_sab$`Molecular Typing`,
                 prefix_len = 2)
```





```r
ggsave('outputs/viz/Figure3_classII.pdf', width = 8, height = 6)

# Performing Eplet Analysis
df <- read.csv("./data/EpletRegistry_ClassII.csv")
df <- calculate_sum(df, OL_ClassII_sab$Molecular.Typing, "OL_unique")
df <- calculate_sum(df, LC_ClassII_sab$Molecular.Typing, "LC_unique")

plot_eplet_overlap(df$OL_unique,
                   df$LC_unique,
                   n1          = "LABScreen",
                   n2          = "Lifecodes",
                   thr         = 1)
```



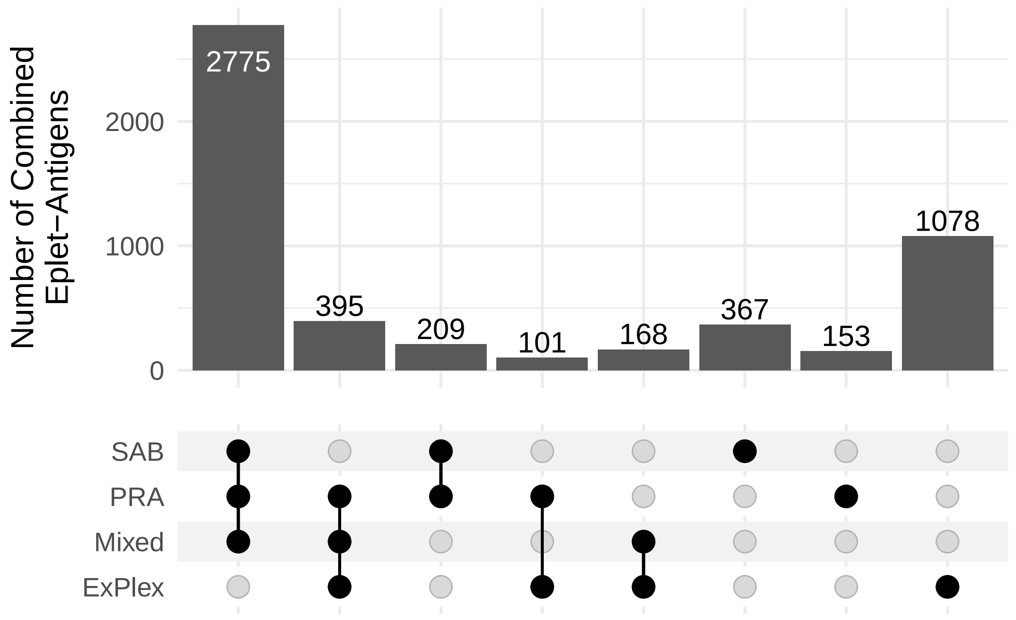
```
ggsave('outputs/viz/Figure3_classII_eplets.pdf', width = 4, height = 4)
```

7 Figure 4: Unique alleles mapped

7.1 Class I

```
# Load and preprocess data
df <- read.csv("./data/EpletRegistry_ClassI.csv")
sab <- process_sab_data(file = "./data/SAB_Class_I_Lot_14_20221212.xlsx",
                        c(1,2))
pra <- process_pra_data(file = "./data/PRA_Class_I_Lot_020_20221212.xlsx", 1)
mix <- process_mix_data(file = "./data/LSM12NC23_024_01.xlsx", 1)
explex <- process_explex_data(file = "./data/ExPlex_classI_II.xlsx", 1)

#Remove non-type-I from Mix
mix <- mix[grepl("A_|B_|C_", mix$values),]

#Making Data Frames
sab.frame <- calculate_sum(df, sab$Molecular.Typing, "SAB_sum",
                           return.sum = FALSE)
sab.frame <- sab.frame%>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1)  %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

explex.frame <- calculate_sum(df, explex$Molecular.Typing, "EXPLEX_sum",
                              return.sum = F)
explex.frame <- explex.frame%>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1)  %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

pra.frame <- calculate_sum(df, pra$values, "PRA_sum",
                           return.sum = F)
pra.frame <- pra.frame%>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1)  %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

mix.frame <- calculate_sum(df, mix$values, "MIX_sum",
                           return.sum = F)
mix.frame <- mix.frame %>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1) %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

b=list(
  SAB = sab.frame,
```

```r
  PRA = pra.frame,
  Mixed = mix.frame,
  ExPlex = explex.frame)

# Create a binary membership matrix
all_elements <- unique(unlist(b))
membership_matrix <- data.frame(
  element = all_elements,
  SAB = all_elements %in% b$SAB,
  PRA = all_elements %in% b$PRA,
  Mixed = all_elements %in% b$Mix,
  ExPlex = all_elements %in% b$ExPlex
)


# Convert logical values to binary
membership_matrix <- membership_matrix %>%
  select(element, ExPlex, Mixed, PRA, SAB)

# Generate the UpSet plot
upset_data <- membership_matrix %>% select(-element)

upset_data <- upset_data[, c("ExPlex", "Mixed", "PRA", "SAB")]

# Then specify the same order in the `intersect` argument:
upset(
  upset_data,
  intersect = colnames(upset_data), # Now in alphabetical order
  base_annotations =
    list('Number of Combined\nEplet-Antigens' = intersection_size()),
  sort_intersections_by='degree',
  themes = upset_default_themes(text = element_text(size = 12)),
  set_sizes = FALSE,
  sort_sets = FALSE
) +
  theme(axis.title.x = element_blank())
```

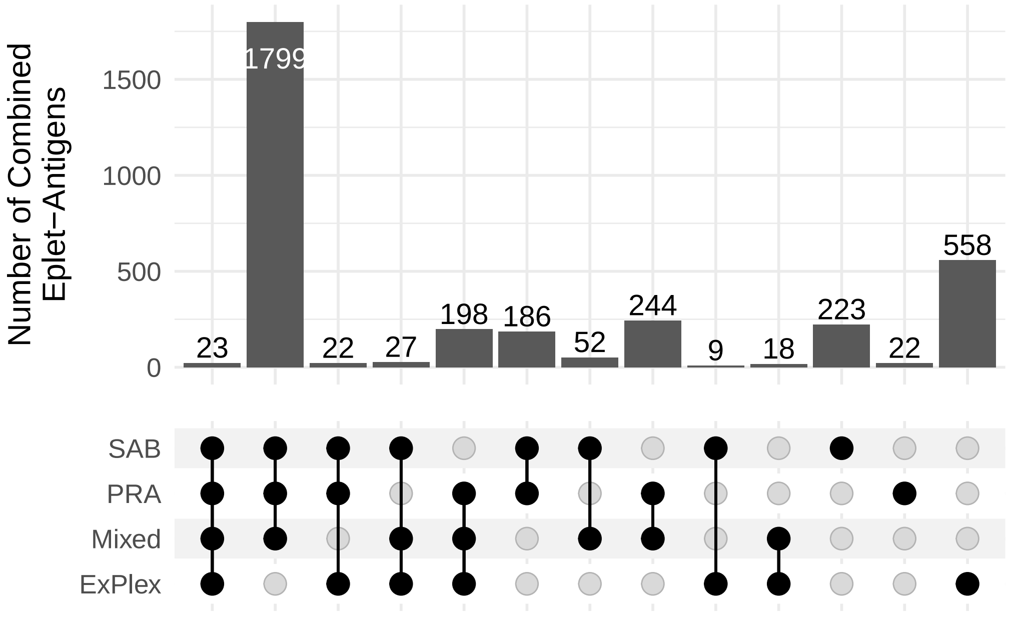


```
ggsave('outputs/viz/Figure4_classI.pdf', width = 8, height = 6)
```

7.2 Class II

```
df <- read.csv("./data/EpletRegistry_ClassII.csv")
sab <- process_sab_data(file = "./data/SAB_ClassII_Lot_15_20221212.xlsx",
                       c(1:2))[-c(1:2),]
pra <- process_pra_data(file = "./data/PRA_Class_II_Lot_019_20221212.xlsx", 1)
mix <- process_mix_data(file = "./data/LSM12NC23_024_01.xlsx", 1)
explex <- process_explex_data(file = "./data/ExPlex_classI_II.xlsx", 2)

#Remove non-type-I from Mix
mix <- mix[-grepl("A_|B_|C_", mix$values),]


#Making Data Frames
sab.frame <- calculate_sum(df, sab$Molecular.Typing, "SAB_sum",
                           return.sum = FALSE)
sab.frame <- sab.frame%>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1)  %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

explex.frame <- calculate_sum(df, explex$Molecular.Typing, "EXPLEX_sum",
                              return.sum = F)
explex.frame <- explex.frame%>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1)  %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

pra.frame <- calculate_sum(df, pra$values, "PRA_sum",
                           return.sum = F)
pra.frame <- pra.frame%>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1)  %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

mix.frame <- calculate_sum(df, mix$values, "MIX_sum",
                           return.sum = F)
mix.frame <- mix.frame %>%
                  select(-c(Epitope.Name, description,
                            exposition, Antibody.Reactivity,
                            evidence, Alleles)) %>%
                  as.data.frame() %>%
                  replace_with_colnames_and_col4() %>%
                  select(-1) %>%
                  unlist() %>%
                  unique() %>%
                  .[. != 0]

b=list(
  SAB = sab.frame,
  PRA = pra.frame,
  Mixed = mix.frame,
  ExPlex = explex.frame)
```

```r
# Create a binary membership matrix
all_elements <- unique(unlist(b))
membership_matrix <- data.frame(
  element = all_elements,
  SAB = all_elements %in% b$SAB,
  PRA = all_elements %in% b$PRA,
  Mixed = all_elements %in% b$Mix,
  ExPlex = all_elements %in% b$ExPlex
)


# Convert logical values to binary
membership_matrix <- membership_matrix %>%
  select(element, ExPlex, Mixed, PRA, SAB)

# Generate the UpSet plot
upset_data <- membership_matrix %>% select(-element)

upset_data <- upset_data[, c("ExPlex", "Mixed", "PRA", "SAB")]

# Then specify the same order in the `intersect` argument:
upset(
  upset_data,
  intersect = colnames(upset_data), # Now in alphabetical order
  base_annotations =
    list('Number of Combined\nEplet-Antigens' = intersection_size()),
  sort_intersections_by='degree',
  themes = upset_default_themes(text = element_text(size = 12)),
  set_sizes = FALSE,
  sort_sets = FALSE
) +
  theme(axis.title.x = element_blank())
```



```r
ggsave('outputs/viz/Figure4_classII.pdf', width = 8, height = 6)
```

8 Conclusion

Here is the comprehensive summary and code of the analysis performed. The following package versions may be helpful in recreating the analysis or making your own.

```r
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: aarch64-apple-darwin20
Running under: macOS Sonoma 14.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LA

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Chicago
tzcode source: internal

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods
[8] base
```

```
other attached packages:
 [1] ggVennDiagram_1.5.2 ComplexUpset_1.3.3  VennDiagram_1.7.3
 [4] futile.logger_1.4.3 proustr_0.4.0       hrbrthemes_0.8.7
 [7] forcats_1.0.0       purrr_1.0.4         readr_2.1.5
[10] tibble_3.2.1        ggplot2_3.5.2       tidyverse_2.0.0
[13] openxlsx_4.2.8      readxl_1.4.5        lubridate_1.9.4
[16] venn_1.12           stringr_1.5.1       janitor_2.2.1
[19] tidyr_1.3.1         data.table_1.17.4   dplyr_1.1.4

loaded via a namespace (and not attached):
 [1] gtable_0.3.6            xfun_0.52               tzdb_0.5.0
 [4] vctrs_0.6.5             tools_4.4.1             generics_0.1.4
 [7] pkgconfig_2.0.3         tokenizers_0.3.0        RColorBrewer_1.1-3
[10] lifecycle_1.0.4         compiler_4.4.1          farver_2.1.2
[13] textshaping_1.0.1       snakecase_0.11.1        fontquiver_0.2.1
[16] fontLiberation_0.1.0    htmltools_0.5.8.1       SnowballC_0.7.1
[19] yaml_2.3.10             Rttf2pt1_1.3.12         pillar_1.10.2
[22] extrafontdb_1.0         admisc_0.38             fontBitstreamVera_0.1.1
[25] tidyselect_1.2.1        zip_2.3.3               digest_0.6.37
[28] stringi_1.8.7           labeling_0.4.3          extrafont_0.19
[31] fastmap_1.2.0           colorspace_2.1-1        cli_3.6.5
[34] magrittr_2.0.3          patchwork_1.3.0         utf8_1.2.5
[37] withr_3.0.2             gdtools_0.4.2           scales_1.4.0
[40] timechange_0.3.0        rmarkdown_2.29          lambda.r_1.2.4
[43] cellranger_1.1.0        ragg_1.4.0              hms_1.1.3
[46] evaluate_1.0.3          knitr_1.50              viridisLite_0.4.2
[49] rlang_1.1.6             futile.options_1.0.1    Rcpp_1.0.14
[52] glue_1.8.0              formatR_1.14            attempt_0.3.1
[55] rstudioapi_0.17.1       jsonlite_2.0.0          R6_2.6.1
[58] systemfonts_1.2.3
```