maftools

Tyler

2025-02-05

```
library(maftools)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
IU112 = read.maf(maf="/Users/tylergross/Desktop/PDX_Variants/maf2/IU112_S101.filtered.annotated.maf.gz")
```

```
## -Reading
## -Validating
## --Non MAF specific values in Variant_Classification column:
##   DE_NOVO_START_IN_FRAME
##   START_CODON_SNP
## -Silent variants: 9815
## -Summarizing
## -Processing clinical data
## --Missing clinical data
## -Finished in 0.842s elapsed (2.575s cpu)
```

```
IU113 = read.maf(maf="/Users/tylergross/Desktop/PDX_Variants/maf2/IU113_S102.filtered.annotated.maf.gz")
```

```
## -Reading
## -Validating
## --Non MAF specific values in Variant_Classification column:
##   COULD_NOT_DETERMINE
##   DE_NOVO_START_IN_FRAME
##   START_CODON_SNP
##   DE_NOVO_START_OUT_FRAME
## -Silent variants: 1634
## -Summarizing
## -Processing clinical data
## --Missing clinical data
## -Finished in 0.186s elapsed (0.702s cpu)
```

```
IU118 = read.maf(maf="/Users/tylergross/Desktop/PDX_Variants/maf2/IU118_S103.filtered.an
notated.maf.gz")
```

```
## -Reading
## -Validating
## --Non MAF specific values in Variant_Classification column:
##   START_CODON_SNP
##   DE_NOVO_START_IN_FRAME
## -Silent variants: 14619
## -Summarizing
## -Processing clinical data
## --Missing clinical data
## -Finished in 1.066s elapsed (3.593s cpu)
```

```
IU119 = read.maf(maf="/Users/tylergross/Desktop/PDX_Variants/maf2/IU119_S104.filtered.an
notated.maf.gz")
```

```
## -Reading
## -Validating
## --Non MAF specific values in Variant_Classification column:
##   START_CODON_SNP
##   DE_NOVO_START_IN_FRAME
##   DE_NOVO_START_OUT_FRAME
## -Silent variants: 12902
## -Summarizing
## --Possible FLAGS among top ten genes:
##   MUC16
## -Processing clinical data
## --Missing clinical data
## -Finished in 0.550s elapsed (2.040s cpu)
```

```
IU120 = read.maf(maf="/Users/tylergross/Desktop/PDX_Variants/maf2/IU120_S105.filtered.an
notated.maf.gz")
```

```
## -Reading
## -Validating
## --Non MAF specific values in Variant_Classification column:
##   DE_NOVO_START_OUT_FRAME
## -Silent variants: 12731
## -Summarizing
## -Processing clinical data
## --Missing clinical data
## -Finished in 0.874s elapsed (2.581s cpu)
```

```
IU121 = read.maf(maf="/Users/tylergross/Desktop/PDX_Variants/maf2/IU121_S106.filtered.an
notated.maf.gz")
```

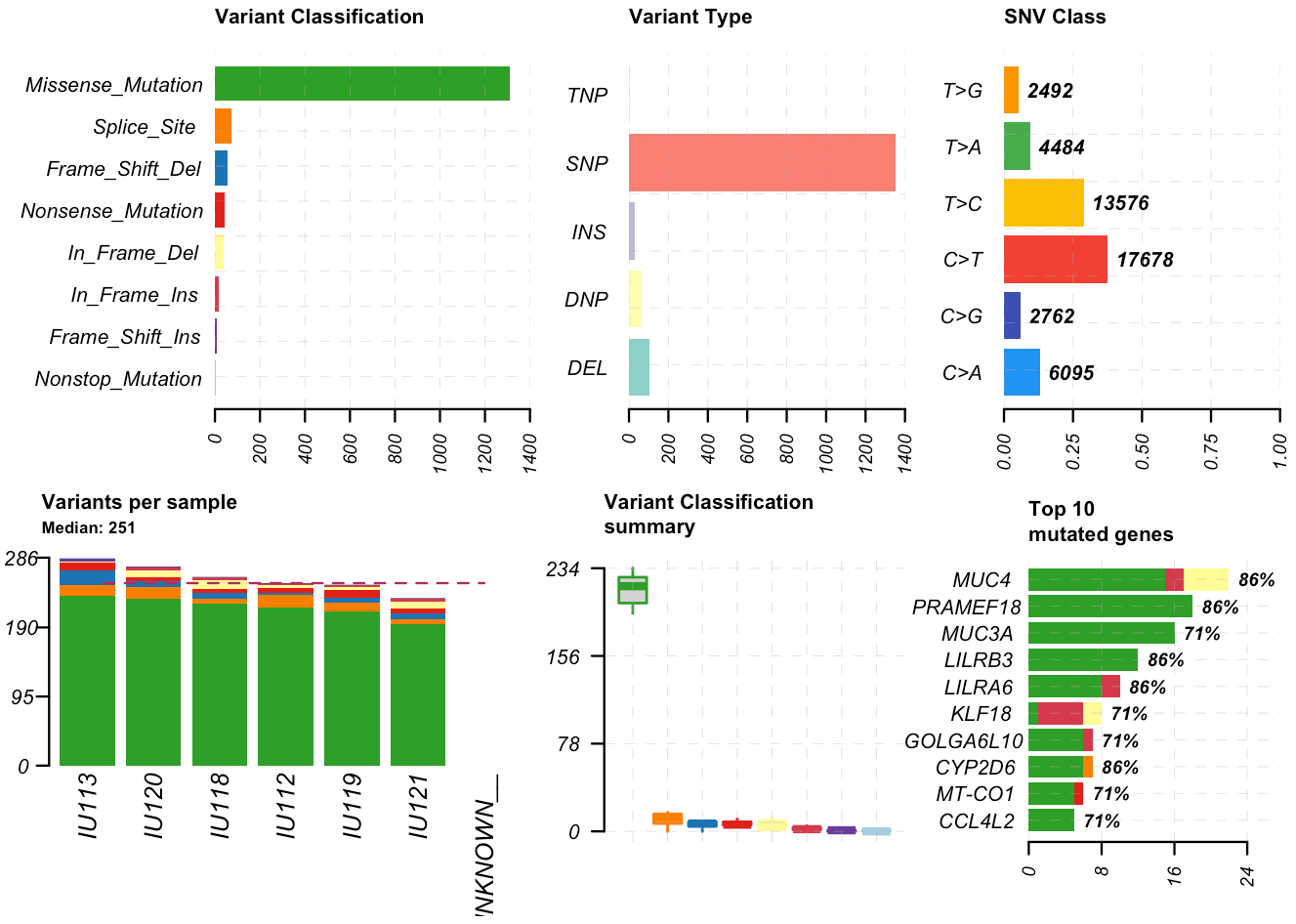
```
## -Reading
## -Validating
## --Non MAF specific values in Variant_Classification column:
##   START_CODON_SNP
##   DE_NOVO_START_OUT_FRAME
##   COULD_NOT_DETERMINE
## -Silent variants: 2295
## -Summarizing
## -Processing clinical data
## --Missing clinical data
## -Finished in 0.219s elapsed (0.801s cpu)
```

```
# Manually assign Tumor_Sample_Barcode to each MAF object
IU112@data$Tumor_Sample_Barcode <- "IU112"
IU113@data$Tumor_Sample_Barcode <- "IU113"
IU118@data$Tumor_Sample_Barcode <- "IU118"
IU119@data$Tumor_Sample_Barcode <- "IU119"
IU120@data$Tumor_Sample_Barcode <- "IU120"
IU121@data$Tumor_Sample_Barcode <- "IU121"

# Create a list of MAF data frames
maf_list <- list(IU112, IU113, IU118, IU119, IU120, IU121)

# You can now use this list as input for merge_mafs
IU_maf <- merge_mafs(mafs = maf_list, verbose = TRUE)
```

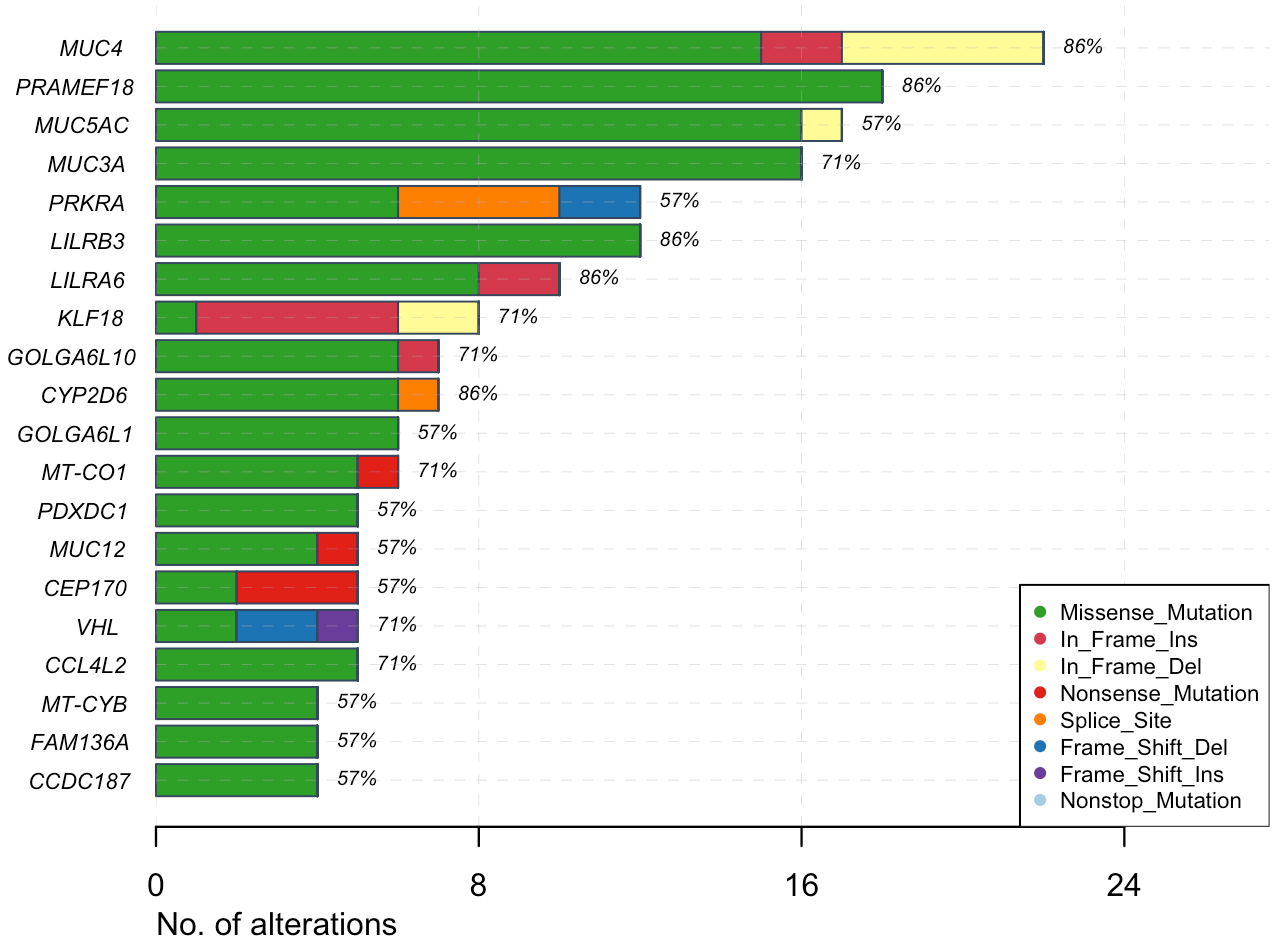
```
## Merging 6 MAF objects
## -Validating
## --Removed 2564 duplicated variants
## --Non MAF specific values in Variant_Classification column:
##   DE_NOVO_START_IN_FRAME
##   START_CODON_SNP
##   COULD_NOT_DETERMINE
##   DE_NOVO_START_OUT_FRAME
## -Silent variants: 51432
## -Summarizing
## -Processing clinical data
## --Annotation missing for below samples in MAF:
##   IU112
##   IU113
##   IU118
##   IU119
##   IU120
##   IU121
## -Finished in 1.959s elapsed (3.655s cpu)
```

```
IU_maf@data <- IU_maf@data[IU_maf@data$Tumor_Sample_Barcode != "__UNKNOWN__", ]
```

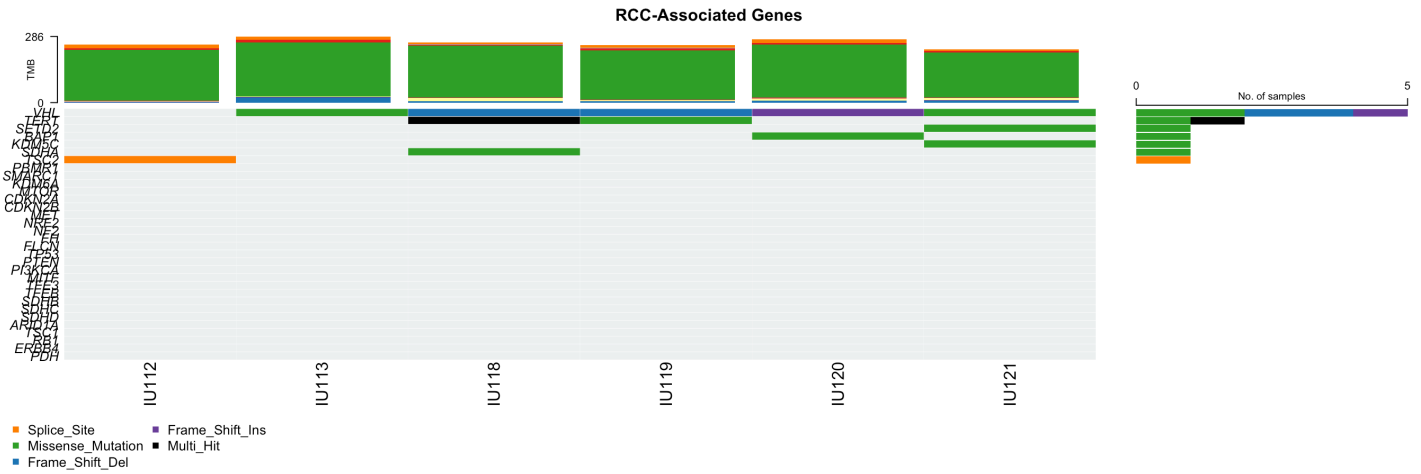
```
plotmafSummary(maf=IU_maf, rmOutlier = TRUE, addStat = 'median', dashboard = TRUE, titvRaw = FALSE, showBarcodes = TRUE)
```



```
mafbarplot(maf=IU_maf,n=20)
```



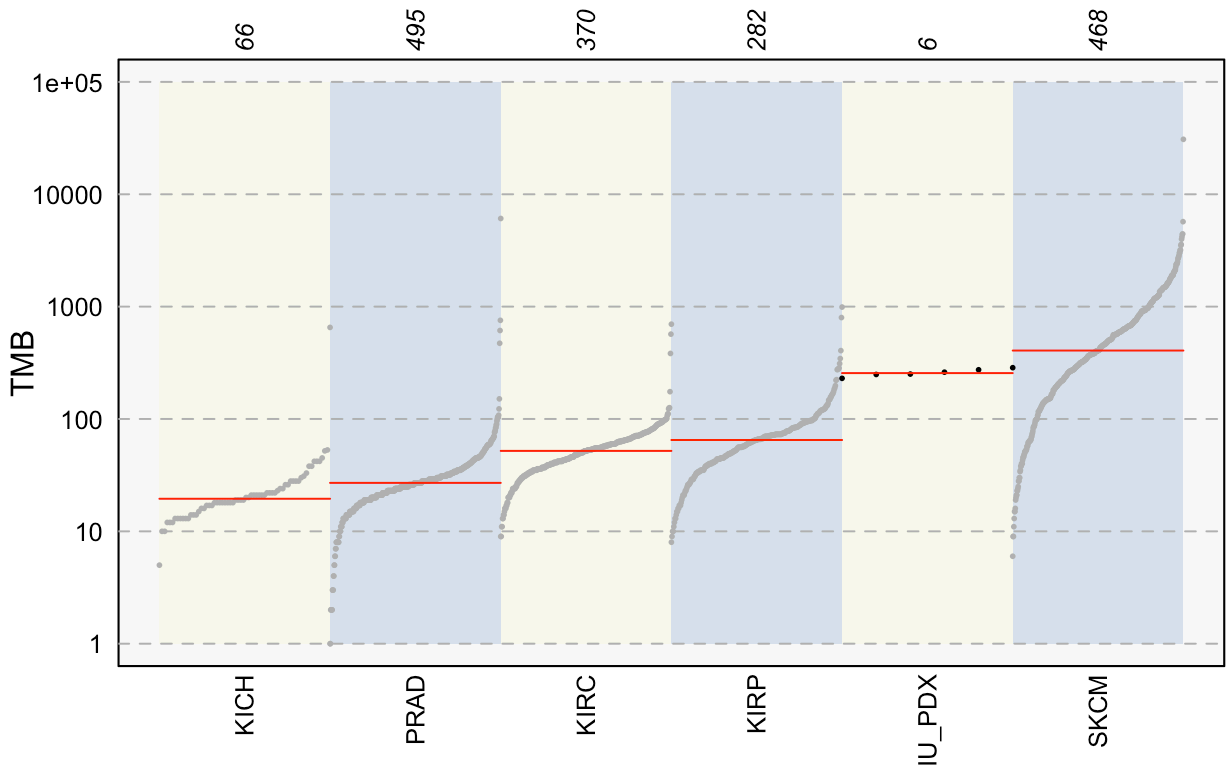

```
oncoplot(maf=IU_maf, genes = c("VHL", "PBMR1", "SETD2", "BAP1", "SMARC1", "KDM5C", "KDM6
A", "MTOR",
           "CDKN2A", "CDKN2B", "MET", "NRF2", "NF2", "FH", "FLCN", "TP53", "TERT",
           "PTEN", "PI3KCA", "MITF", "TFE3", "TFEB", "SDHA", "SDHB", "SDHC", "SDHD",
           "ARID1A", "TSC1", "TSC2", "RB1", "ERBB4", "PDH"), showTumorSampleBarcodes =
TRUE, sampleOrder = c("IU112","IU113","IU118","IU119","IU120","IU121"), SampleNamefontSi
ze = 1.5, titleText = "RCC-Associated Genes", showPct = FALSE)
```



```
laml.mutload = tcgaCompare(maf = IU_maf, cohortName = 'IU_PDX', logscale = TRUE, tcga_co
horts = c('KIRC','KICH','KIRP','PRAD','SKCM'))
```

```
## Warning in FUN(X[[i]], ...): Removed 1 samples with zero mutations.
```

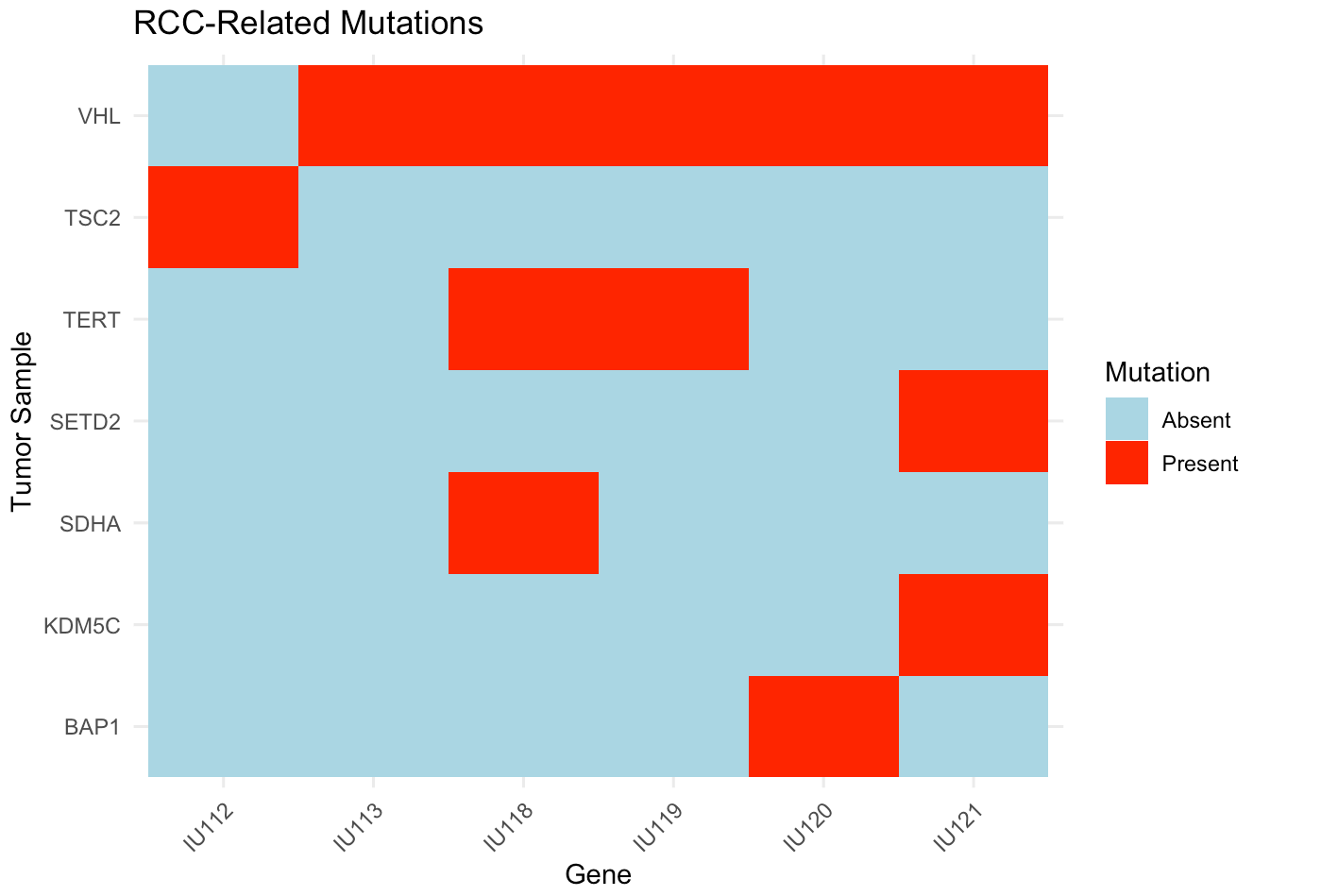
```
## Performing pairwise t-test for differences in mutation burden..
```




```
# List of Hugo Symbols you want to filter by
## from https://pmc.ncbi.nlm.nih.gov/articles/PMC7459851/
hugo_symbols <- c("VHL", "PBMR1", "SETD2", "BAP1", "SMARC1", "KDM5C", "KDM6A", "MTOR",
                  "CDKN2A", "CDKN2B", "MET", "NRF2", "NF2", "FH", "FLCN", "TP53", "TER
T",
                  "PTEN", "PI3KCA", "MITF", "TFE3", "TFEB", "SDHA", "SDHB", "SDHC", "SDH
D",
                  "ARID1A", "TSC1", "TSC2", "RB1", "ERBB4", "PDH")

# Filter IU_maf@data for rows where Hugo_Symbol is in the list of symbols
RCC_genes <- IU_maf@data %>% dplyr::filter(Hugo_Symbol %in% hugo_symbols)

# View the filtered data
RCC_genes = RCC_genes %>% dplyr::select(Hugo_Symbol, Variant_Classification, Variant_Typ
e, Tumor_Sample_Barcode, Genome_Change, Protein_Change)
print(RCC_genes)
```

```
## Index: <Hugo_Symbol>
##     Hugo_Symbol Variant_Classification Variant_Type Tumor_Sample_Barcode
##           <char>                 <fctr>       <fctr>               <fctr>
##  1:        TSC2           Splice_Site          DEL                IU112
##  2:         VHL      Missense_Mutation          SNP                IU113
##  3:         VHL        Frame_Shift_Del          DEL                IU118
##  4:        SDHA      Missense_Mutation          SNP                IU118
##  5:        TERT      Missense_Mutation          SNP                IU118
##  6:        TERT      Missense_Mutation          SNP                IU118
##  7:         VHL        Frame_Shift_Del          DEL                IU119
##  8:        TERT      Missense_Mutation          SNP                IU119
##  9:         VHL        Frame_Shift_Ins          INS                IU120
## 10:        BAP1      Missense_Mutation          SNP                IU120
## 11:         VHL      Missense_Mutation          SNP                IU121
## 12:       SETD2      Missense_Mutation          SNP                IU121
## 13:       KDM5C      Missense_Mutation          SNP                IU121
##                                  Genome_Change Protein_Change
##                                         <char>         <char>
##  1: g.chr16:2076166_2076182delCTAAGGTGGGCTCAGGG
##  2:                          g.chr3:10149808G>A        p.C162Y
##  3:                         g.chr3:10146585delC       p.P138fs
##  4:                            g.chr5:225429A>G        p.N108S
##  5:                           g.chr5:1282562G>A        p.H546Y
##  6:                           g.chr5:1294233G>A        p.A218V
##  7:                         g.chr3:10146585delC       p.P138fs
##  8:                           g.chr5:1294233G>A        p.A218V
##  9:                 g.chr3:10146578_10146579insT       p.V137fs
## 10:                          g.chr3:52408061C>T         p.C91Y
## 11:                          g.chr3:10142113T>A         p.L89H
## 12:                          g.chr3:47083732C>A       p.W2016C
## 13:                          g.chrX:53201910C>T        p.G604S
```

```
library(ggplot2)
library(dplyr)
library(tidyr)

# Filter and create binary matrix
mutation_matrix <- RCC_genes %>%
  select(Hugo_Symbol, Tumor_Sample_Barcode) %>%
  distinct() %>%  # Remove duplicate mutations of the same gene in the same sample
  mutate(Present = 1) %>%
  pivot_wider(names_from = Hugo_Symbol, values_from = Present, values_fill = list(Presen
t = 0))

# Convert to long format for ggplot
mutation_long <- mutation_matrix %>%
  pivot_longer(cols = -Tumor_Sample_Barcode, names_to = "Gene", values_to = "Mutation")

# Plot the heatmap
ggplot(mutation_long, aes(x = Tumor_Sample_Barcode, y = Gene, fill = factor(Mutation)))
+
  geom_tile() +
  scale_fill_manual(values = c("lightblue", "red"), name = "Mutation", labels = c("Absen
t", "Present")) +
  theme_minimal() +
  labs(title = "RCC-Related Mutations", x = "Gene", y = "Tumor Sample") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# List of Hugo Symbols you want to filter by
## from supplementary table 4 of https://www.nature.com/articles/s41375-022-01804-w#MOESM4
EBV_hugo_symbols <- c("SOCS1","CD58","NOTCH2","NOTCH1","B2M","FOX01", "MTOR", "KMT2C","ARID1A","TP53")

# Filter IU_maf@data for rows where Hugo_Symbol is in the list of symbols
EBV_genes <- IU_maf@data %>% dplyr::filter(Hugo_Symbol %in% EBV_hugo_symbols)
```

```
VHL_loll=lollipopPlot(maf=IU_maf, gene='VHL', AACol = 'Protein_Change', showMutationRate = FALSE, labelPos = "all")
```

```
## 2 transcripts available. Use arguments refSeqID or proteinID to manually specify tx name.
```

```
##      HGNC refseq.ID protein.ID aa.length
##    <char>    <char>     <char>     <num>
## 1:    VHL NM_000551  NP_000542       213
## 2:    VHL NM_198156  NP_937799       172
```

```
## Using longer transcript NM_000551 for now.
```