

MiRUtil examples

mikessh

2018-07-17

Install package via devtools:

```
#devtools::install_github("milaboratory/mirutil")
#library(mirutil); packageVersion("mirutil")
#devtools::unload()
#devtools::document()
devtools::load_all()
```

```
## Loading mirutil
## Loading required package: MASS
## Loading required package: magrittr
## Loading required package: data.table
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:data.table':
##
##   between, first, last
## The following object is masked from 'package:MASS':
##
##   select
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
## Loading required package: stringr
## Loading required package: reshape2
##
## Attaching package: 'reshape2'
## The following objects are masked from 'package:data.table':
##
##   dcast, melt
## Loading required package: parallel
## Loading required package: tidyr
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:reshape2':
```

```
##
```

```
## smiths
```

```
## The following object is masked from 'package:magrittr':
```

```
##
```

```
## extract
```

Load metadata and samples

```
metadata <- fread("metadata.txt")
dataset <- metadata %>% read_mixcr_dataset
```

Compute rearrangement statistics

```
library(kableExtra) # for tables
```

```
df.stats <- dataset %>%
  compute_rearr_stats
```

```
df.stats %>%
  str
```

```
## List of 3
```

```
## $ segment.usage:Classes 'data.table' and 'data.frame': 708 obs. of 8 variables:
```

```
## ..$ segment : chr [1:708] "TRAV1-1" "TRAV1-2" "TRAV10" "TRAV11" ...
```

```
## ..$ segment.type : chr [1:708] "V" "V" "V" "V" ...
```

```
## ..$ count.clonotypes: num [1:708] 276.128 170.698 250 0.578 557.852 ...
```

```
## ..$ count.reads : num [1:708] 286.128 189.524 263 0.578 594.852 ...
```

```
## ..$ freq.clonotypes : num [1:708] 1.84e-02 1.14e-02 1.67e-02 3.86e-05 3.72e-02 ...
```

```
## ..$ freq.reads : num [1:708] 1.76e-02 1.17e-02 1.62e-02 3.56e-05 3.66e-02 ...
```

```
## ..$ sample.id : chr [1:708] "2U.B.TRA.1" "2U.B.TRA.1" "2U.B.TRA.1" "2U.B.TRA.1" ...
```

```
## ..$ chain : chr [1:708] "TRA" "TRA" "TRA" "TRA" ...
```

```
## ..- attr(*, ".internal.selfref")=<externalptr>
```

```
## $ insert.size :Classes 'data.table' and 'data.frame': 505 obs. of 8 variables:
```

```
## ..$ insertions : num [1:505] 0 1 2 3 4 5 6 7 8 9 ...
```

```
## ..$ ins.type : chr [1:505] "VJ" "VJ" "VJ" "VJ" ...
```

```
## ..$ count.clonotypes: int [1:505] 2701 1175 1620 1862 1890 1393 1061 845 621 473 ...
```

```
## ..$ count.reads : int [1:505] 3120 1253 1757 2042 1996 1465 1123 881 714 509 ...
```

```
## ..$ freq.clonotypes : num [1:505] 0.1803 0.0784 0.1081 0.1243 0.1262 ...
```

```
## ..$ freq.reads : num [1:505] 0.1921 0.0771 0.1082 0.1257 0.1229 ...
```

```
## ..$ sample.id : chr [1:505] "2U.B.TRA.1" "2U.B.TRA.1" "2U.B.TRA.1" "2U.B.TRA.1" ...
```

```
## ..$ chain : chr [1:505] "TRA" "TRA" "TRA" "TRA" ...
```

```
## ..- attr(*, ".internal.selfref")=<externalptr>
```

```
## $ deletion.size:Classes 'data.table' and 'data.frame': 10926 obs. of 9 variables:
```

```
## ..$ segment : chr [1:10926] "TRAV1-1" "TRAV1-1" "TRAV1-1" "TRAV1-1" ...
```

```
## ..$ deletions : int [1:10926] -13 -11 -10 -9 -8 -7 -6 -5 -4 -3 ...
```

```
## ..$ del.type : chr [1:10926] "V3" "V3" "V3" "V3" ...
```

```
## ..$ count.clonotypes: int [1:10926] 3 2 1 8 20 16 23 18 24 28 ...
```

```
## ..$ count.reads : int [1:10926] 3 2 1 8 20 16 23 20 25 28 ...
```

```
## ..$ freq.clonotypes : num [1:10926] 2.00e-04 1.33e-04 6.67e-05 5.34e-04 1.33e-03 ...
```

```
## ..$ freq.reads : num [1:10926] 1.85e-04 1.23e-04 6.16e-05 4.92e-04 1.23e-03 ...
```

```
## ..$ sample.id : chr [1:10926] "2U.B.TRA.1" "2U.B.TRA.1" "2U.B.TRA.1" "2U.B.TRA.1" ...
```

```
## ..$ chain : chr [1:10926] "TRA" "TRA" "TRA" "TRA" ...
```

```
## ..- attr(*, ".internal.selfref")=<externalptr>
```

Plot segment usage for alpha chain

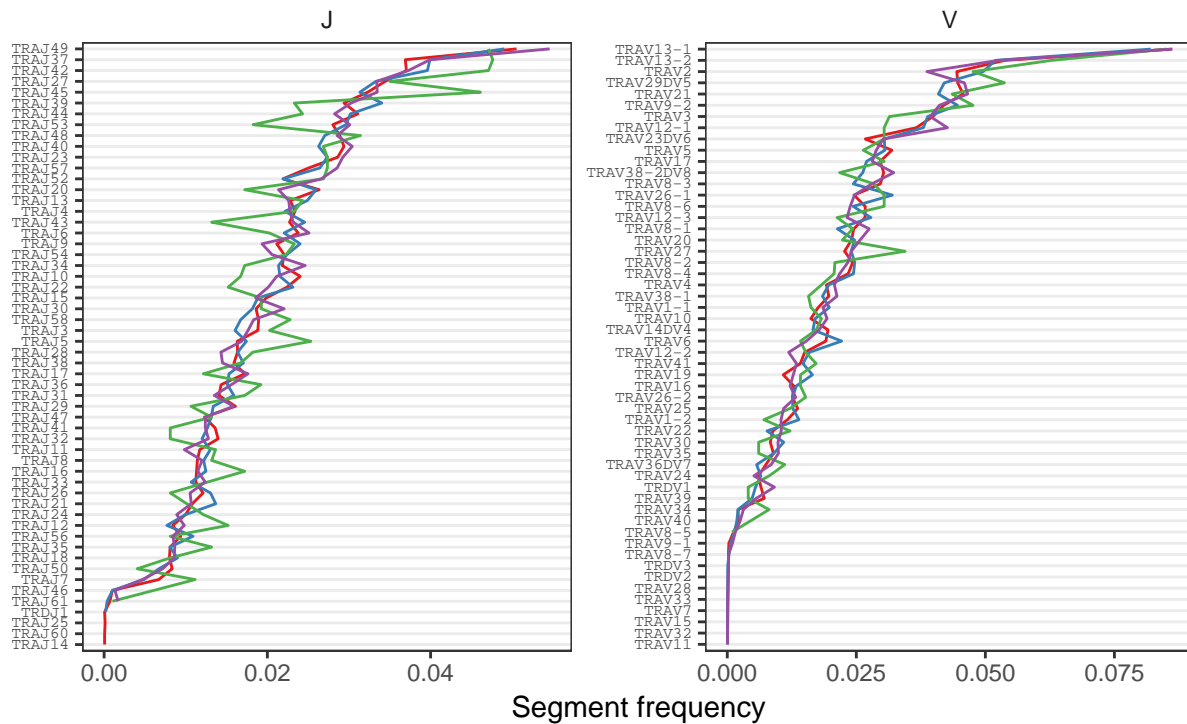
```

# packages for plotting
library(ggplot2)
library(forcats)

plt_segm <- function(m) {
  m %>%
    ggplot(aes(x = fct_reorder(segment, freq.reads),
               group = sample.id,
               y = freq.reads, color = sample.id)) +
    geom_line() +
    coord_flip() +
    xlab("") + ylab("Segment frequency") +
    scale_color_brewer(palette = "Set1") +
    facet_wrap(~segment.type, scales = "free") +
    theme_bw() +
    theme(legend.position = "bottom",
          axis.text.y = element_text(family = "mono", size = 6),
          strip.background = element_rect(fill = NA, color = NA),
          panel.grid.major.x = element_blank(),
          panel.grid.minor = element_blank())
}

df.stats %>%
  .$segment.usage %>%
  filter(chain == "TRA") %>%
  plt_segm

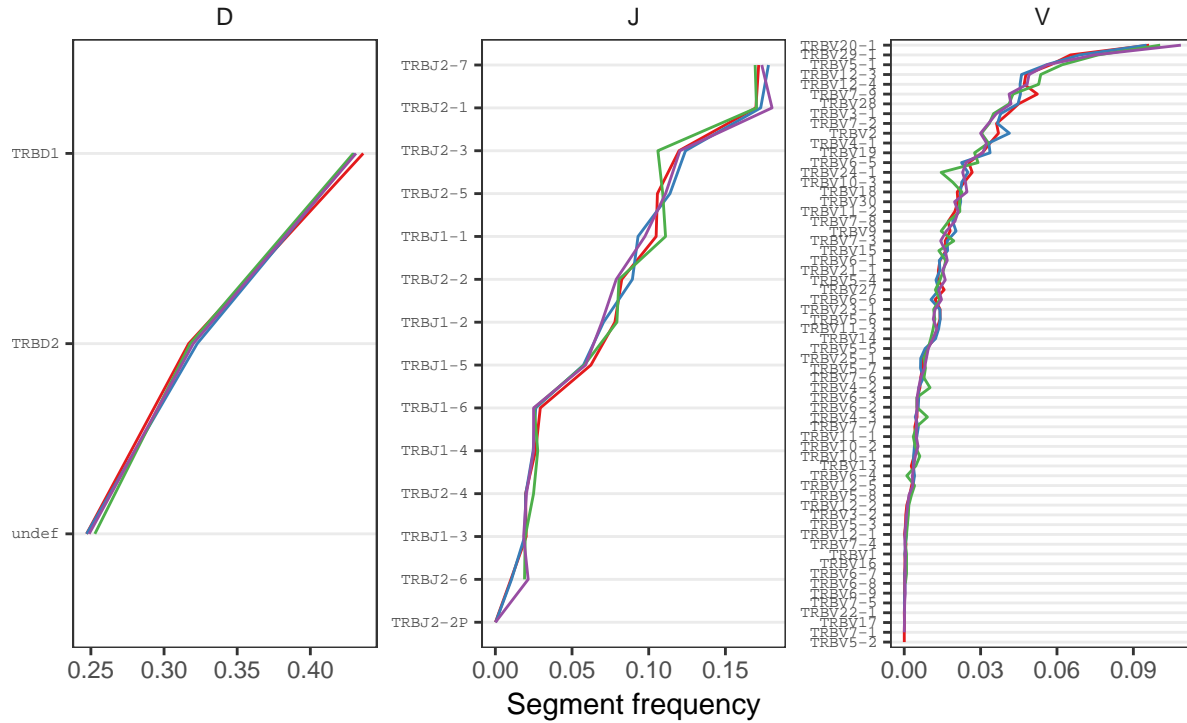
```



sample.id — 2U.B.TRA.1 — 2U.B.TRA.2 — U.B.TRA.1 — U.B.TRA.2

For beta chain

```
df.stats %>%
  .$segment.usage %>%
  filter(chain == "TRB") %>%
  plt_segm
```



sample.id — 2U.B.TRB.1 — 2U.B.TRB.2 — U.B.TRB.1 — U.B.TRB.2

Get distances between samples based on rearrangement statistics

```
df.dist <- df.stats %>%
  compute_rearr_stat_dist

df.dist %>%
  head
```

##	d	sample.id.1	sample.id.2	type	chain	value.type	statistic
## 1:	0.05167906	2U.B.TRA.2	2U.B.TRA.1	V	TRA	reads	segment.usage
## 2:	0.09695837	U.B.TRA.1	2U.B.TRA.1	V	TRA	reads	segment.usage
## 3:	0.04813716	U.B.TRA.2	2U.B.TRA.1	V	TRA	reads	segment.usage
## 4:	0.10223805	U.B.TRA.1	2U.B.TRA.2	V	TRA	reads	segment.usage
## 5:	0.06709979	U.B.TRA.2	2U.B.TRA.2	V	TRA	reads	segment.usage
## 6:	0.03982001	2U.B.TRB.2	2U.B.TRB.1	V	TRB	reads	segment.usage

Convert distances to MDS coordinates

```
df.mds <- df.dist %>% compute_rearr_stat_mds
```

```
## initial value 0.472954
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
```

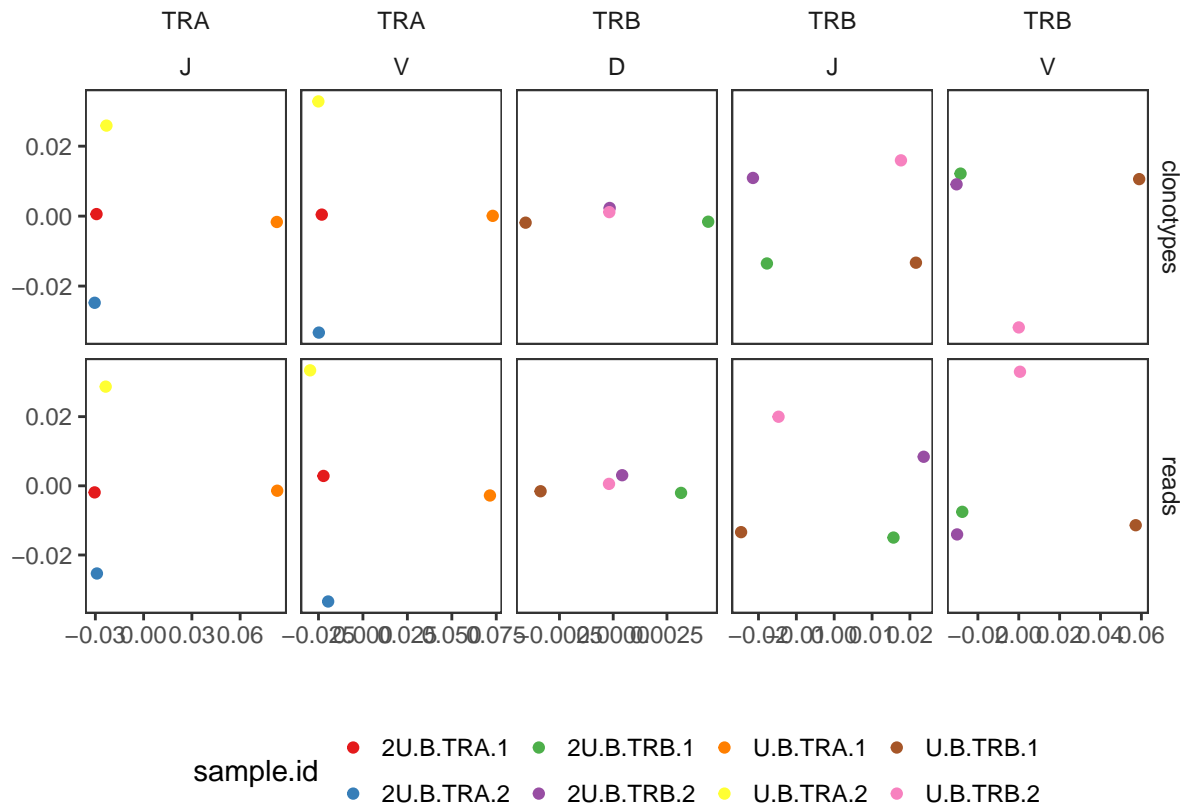
```
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.943065
## final value 0.000000
## converged
## initial value 0.803030
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.061454
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.273251
## final value 0.000000
## converged
## initial value 0.562183
## final value 0.000000
## converged
## initial value 0.777526
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
```

```
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
## initial value 0.000000
## final value 0.000000
## converged
```

Plot MDS for segment usage

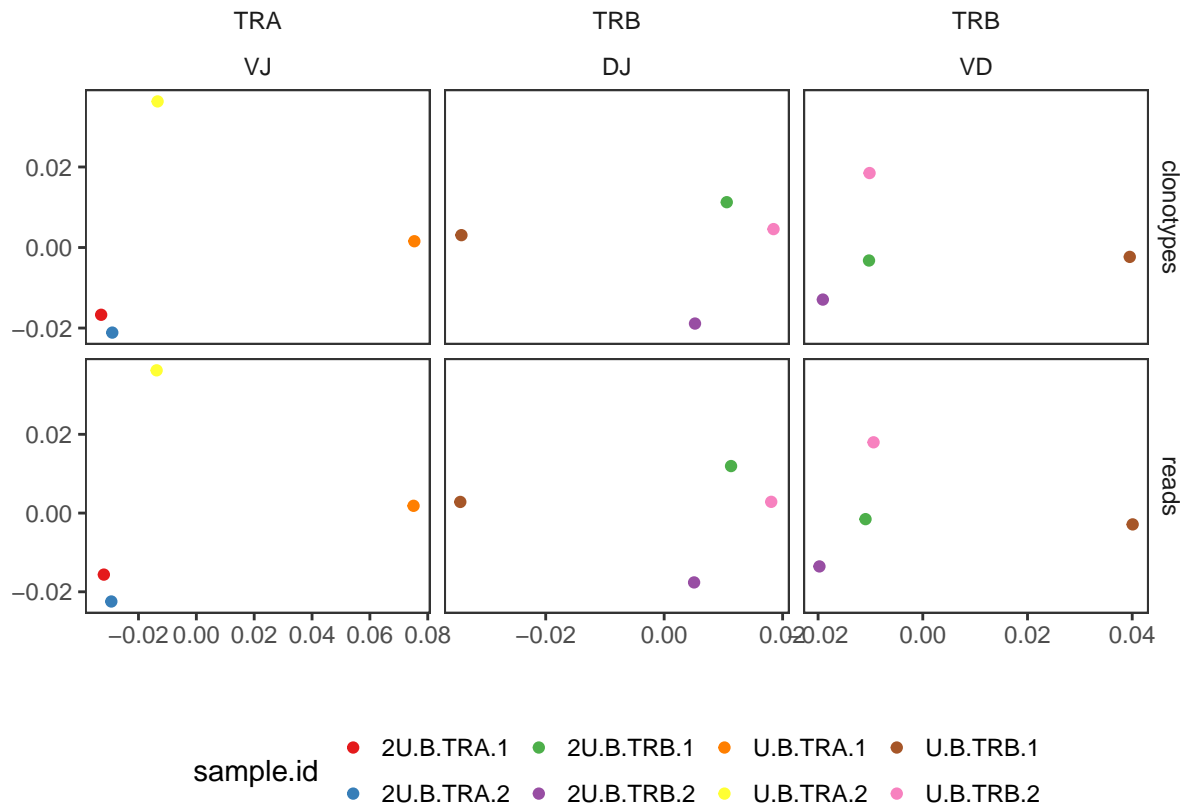
```
plt_mds <- function(m) {
  m %>%
    ggplot(aes(x = mds.x, y = mds.y, color = sample.id)) +
    geom_point() +
    xlab("") + ylab("") +
    facet_grid(value.type~chain + type, scales = "free") +
    scale_color_brewer(palette = "Set1") +
    theme_bw() +
    theme(legend.position = "bottom",
          strip.background = element_rect(fill = NA, color = NA),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
}

df.mds %>%
  filter(statistic == "segment.usage") %>%
  plt_mds
```



Plot MDS for added N-bases (inserts)

```
df.mds %>%
  filter(statistic == "insert.size") %>%
  plt_mds
```



Plot MDS for trimmed bases from segment ends

```
df.mds %>%
  filter(statistic == "deletion.size") %>%
  plt_mds
```