Analyzing public and MLANA-specific clonotypes in TCR repertoires extracted from TCGA RNA-Seq data

Looking at public clonotypes in TCGA - clonotypes found in two+ donors have shorter insert size, suggesting a canonical mechanism related to convergent recombination

```
library(ggplot2)
library(reshape2)
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
```

```
library(survminer)
```

```
## Loading required package: ggpubr
```

```
##
## Attaching package: 'survminer'
```

```
## The following object is masked from 'package:ggpubr':
##
##     theme_classic2
```

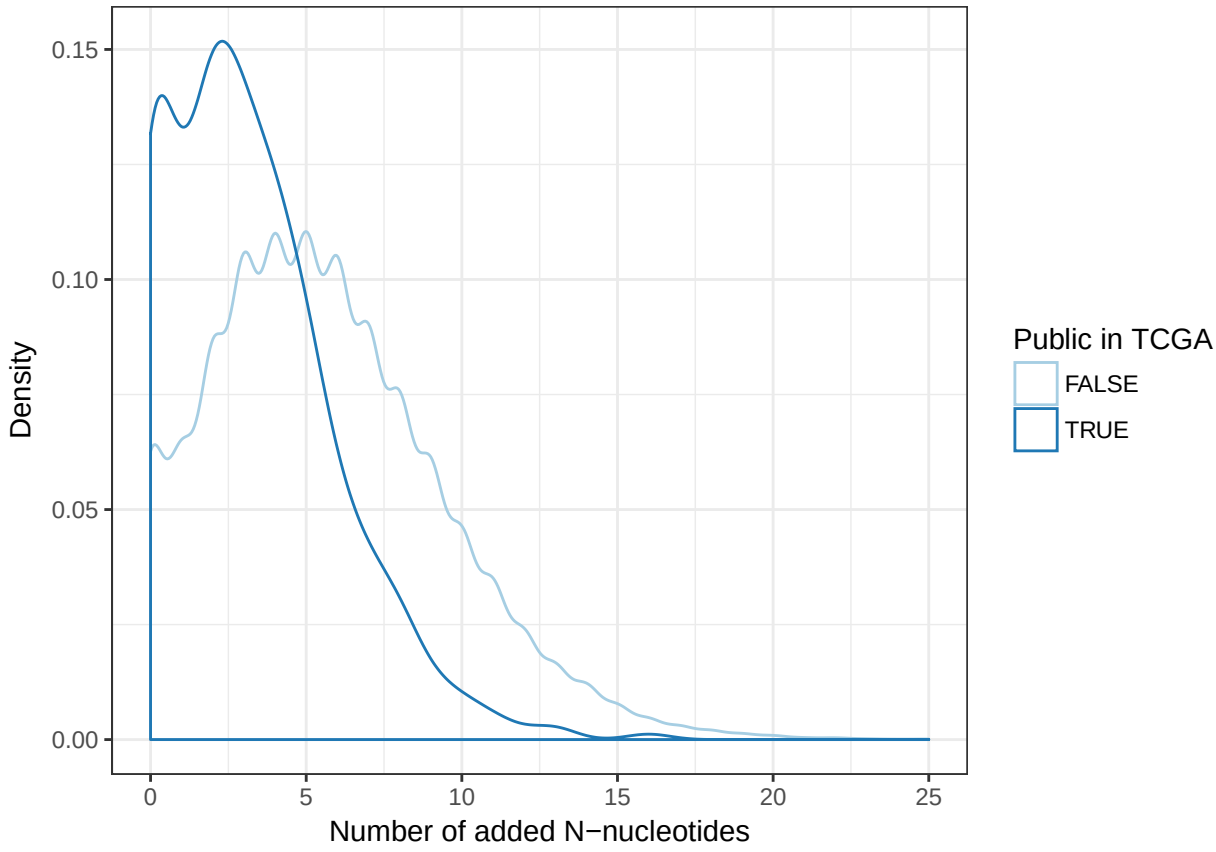
```
## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
library(survival)

df = read.table("pooled_tcga_samples.txt", header=T, sep="\t") %>%
  filter(startsWith(as.character(v), "TRB"))

df$insert.size = pmax(0, pmin(25, with(df, ifelse(DStart<0, JStart-VEnd - 1, DStart-VEnd+JStart-DEnd - 1
df$public_tcga = df$incidence > 1

ggplot(df, aes(insert.size, color=public_tcga)) +
  geom_density() +
  scale_color_brewer("Public in TCGA", palette = "Paired") +
  xlab("Number of added N-nucleotides") + ylab("Density") +
  theme_bw()
```



Testing relationship between being public in TCGA and matching one of 10000 known public clonotypes or one of 3209 MLANA-specific clonotypes

```
publics = read.table("publics.txt")
mlana = read.table("mlana.txt")

df$public = df$cdr3aa %in% publics$V1
df$mlana = df$cdr3aa %in% mlana$V1

df.1 = df %>%
  group_by(public_tcga, public) %>%
  summarize(count = n())

m1 = dcast(df.1, public_tcga ~ public)
```

```
## Using count as value column: use value.var to override.
```

```
colnames(m1) = paste("public", colnames(m1), sep ="_")
rownames(m1) = paste("public_tcga", m1[,1], sep ="_")
m1[,1] = NULL
print(m1)
```

```
##                   public_FALSE public_TRUE
## public_tcga_FALSE        21950         399
## public_tcga_TRUE           454          31
```

```
fisher.test(m1)
```

```
##
```

```
##  Fisher's Exact Test for Count Data
##
## data:  m1
## p-value = 3.79e-09
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  2.488589 5.491703
## sample estimates:
## odds ratio
##   3.756121
```

```
df.2 = df %>%
  group_by(public_tcga, mlana) %>%
  summarize(count = n())

m2 = dcast(df.2, public_tcga ~ mlana)
```

```
## Using count as value column: use value.var to override.
```

```
colnames(m2) = paste("mlana", colnames(m2), sep ="_")
rownames(m2) = paste("public_tcga", m2[,1], sep ="_")
m2[,1] = NULL
print(m2)
```

```
##                   mlana_FALSE mlana_TRUE
## public_tcga_FALSE       22272         77
## public_tcga_TRUE          464         21
```

```
fisher.test(m2)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  m2
## p-value = 1.356e-15
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   7.599347 21.637340
## sample estimates:
## odds ratio
##   13.08257
```

TCGA-public MLANA-specific clonotypes

```
print(subset(df, public_tcga & mlana) %>% dplyr::select(count,cdr3aa,v,j,incidence))
```

```
##      count         cdr3aa        v       j incidence
## 148     20  CASSLGQAYEQYF  TRBV7-8 TRBJ2-7        11
## 227     15    CASSLGYEQYF  TRBV7-6 TRBJ2-7         6
## 246     14   CASSLGNTEAFF   TRBV18 TRBJ1-1         4
## 348     10    CASSFSYEQYF   TRBV13 TRBJ2-7         7
## 434      9   CASSLGRNEQFF  TRBV7-2 TRBJ2-1         2
## 757      6    CASSLGGELFF   TRBV13 TRBJ2-2         4
## 793      6    CASSPGYEQYF   TRBV28 TRBJ2-7         5
## 832      6  CASSLGGGNEQFF   TRBV27 TRBJ2-1         3
## 1388     4 CSVGTGGTNEKLFF TRBV29-1 TRBJ1-4         4
## 1731     3 CASSFLAGTDTQYF   TRBV28 TRBJ2-3         2
```

```
## 1741     3    CASSLNNEQFF  TRBV7-3 TRBJ2-1         2
## 2089     3  CASSLGSTDTQYF   TRBV27 TRBJ2-3         3
## 2287     3    CASRGNTEAFF  TRBV7-9 TRBJ1-1         2
## 2420     3 CASSTGDSNQPQHF  TRBV6-1 TRBJ1-5         2
## 2598     2 CASSLGGGNQPQHF TRBV12-3 TRBJ1-5         2
## 3900     2  CASSLGQNNEQFF TRBV12-3 TRBJ2-1         2
## 4116     2    CASSSPYEQYF   TRBV28 TRBJ2-7         2
## 4838     2 CASSLGGVNTEAFF   TRBV19 TRBJ1-1         2
## 5110     2  CASSLTGTDTQYF  TRBV7-8 TRBJ2-3         2
## 5279     2  CASSEGRSYEQYF  TRBV5-6 TRBJ2-7         2
## 5502     2 CASSLVGSSYEQYF  TRBV7-8 TRBJ2-7         2
```

Per sample annotation using VDJdb-standalone (2 mismatches allowed), appending patient metadata for survival analysis

```
df.annot = read.table("tcga_mlana_annotation_summary.txt", header=T, sep="\t") %>%
  filter(db.column.name == "summary" & db.column.value == "found") %>%
  select(sample_id, reads, unique, frequency)

df.annot.total = read.table("tcga_mlana_annotation_summary.txt", header=T, sep="\t") %>%
  filter(db.column.name == "summary") %>%
  group_by(sample_id) %>%
  summarize(reads_total = sum(reads), unique_total = sum(unique), frequency_total = sum(frequency)) %>%
  select(sample_id, reads_total, unique_total, frequency_total)

df.annot = merge(df.annot, df.annot.total)

df.conv = read.table("TCGA_to_SKCM.csv", header = T)
colnames(df.conv) = c("sample_id", "skcm_id")
df.annot$sample_id = substr(df.annot$sample_id, 1, 12)

df.annot = merge(df.annot, df.conv)

df.meta = read.table("survival.txt", header=F, sep = "\t")
colnames(df.meta) = c("skcm_id", "survival", "status", "age", "stage", "sex")

df.annot = merge(df.annot, df.meta)
df.annot$survival = as.numeric(as.character(df.annot$survival))
```

```
## Warning: NAs introduced by coercion
```

Survival is correlated with total number of TRB reads and the number of TRB reads coming from MLANA-specific clonotypes. Here we set a threshold as the median of TRB reads.

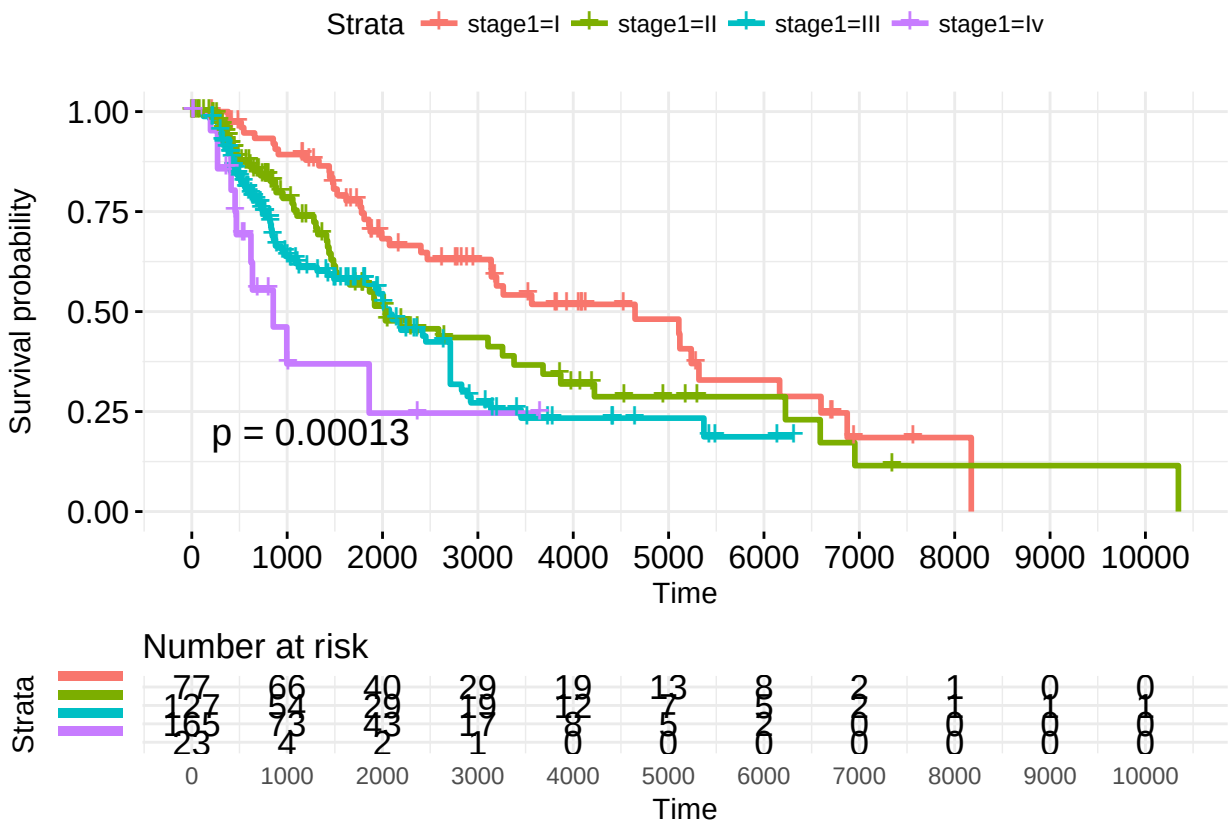
```
df.annot$high_reads_total = with(df.annot, reads_total > median(reads_total))
df.annot$high_reads_mlana = with(df.annot, reads > median(reads))
df.annot$stage1 = with(df.annot, ifelse(stage %in% c("I", "Ia", "Ib"), "I",
                                        ifelse(stage %in% c("II", "IIa", "IIb", "IIc"), "II",
                                               ifelse(stage %in% c("III", "IIIa", "IIIb", "IIIc"), "II
                                                      ifelse(stage == "Iv", "Iv", NA)))))

fit.0 = survfit(Surv(time = survival, event = status == "dead") ~ stage1,
                data = df.annot)
print(fit.0)
```

```
## Call: survfit(formula = Surv(time = survival, event = status == "dead") ~
```

```
##     stage1, data = df.annot)
##
##    58 observations deleted due to missingness
##                n events median 0.95LCL 0.95UCL
## stage1=I     77     39   4648    3139    6164
## stage1=II   128     51   2028    1506    3683
## stage1=III  165     82   2071    1832    2711
## stage1=Iv    23     11    854     620      NA
```

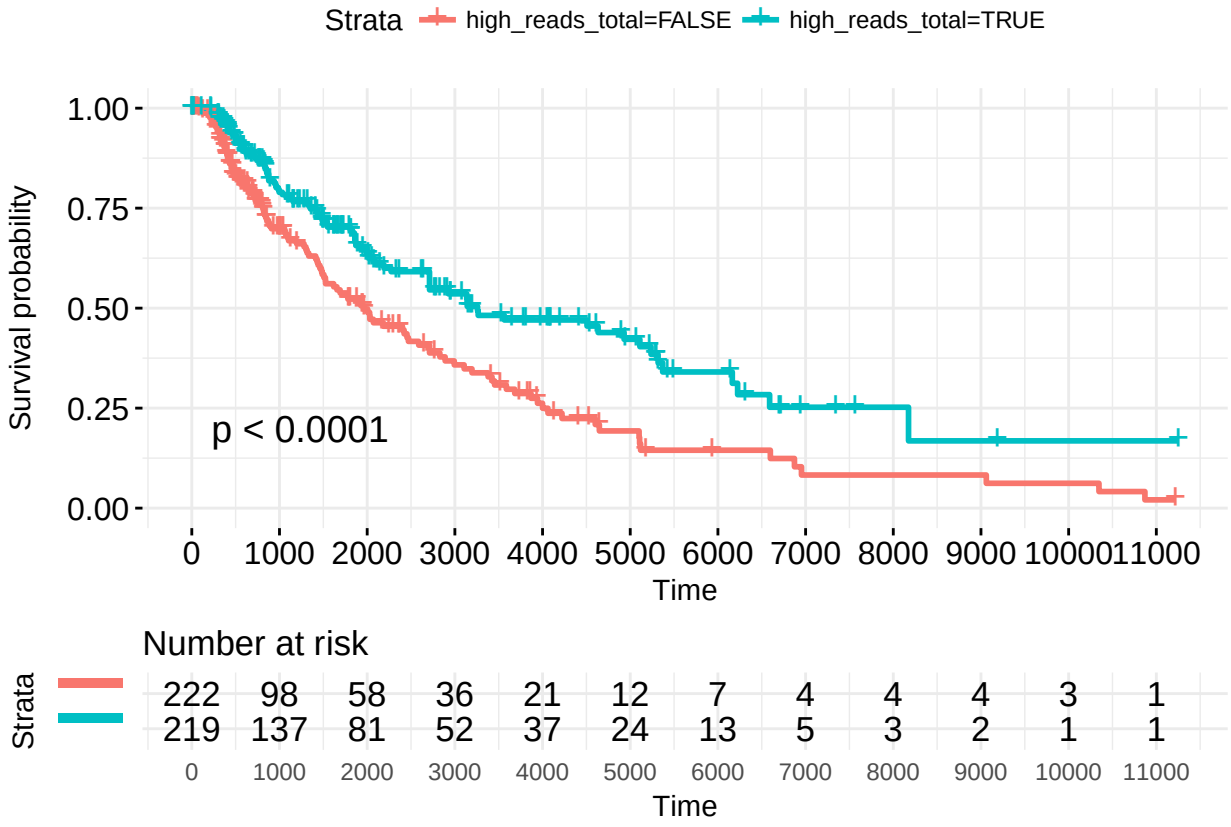
```
ggsurvplot(fit.0, data = df.annot, risk.table = T, pval = T, break.time.by = 1000,
           ggtheme = theme_minimal(), risk.table.y.text.col = T, risk.table.y.text = F)
```




```
fit.1 = survfit(Surv(time = survival, event = status == "dead") ~ high_reads_total,
                data = df.annot)
print(fit.1)
```

```
## Call: survfit(formula = Surv(time = survival, event = status == "dead") ~
##     high_reads_total, data = df.annot)
##
##    9 observations deleted due to missingness
##                            n events median 0.95LCL 0.95UCL
## high_reads_total=FALSE 222    118   1960    1524    2711
## high_reads_total=TRUE  220     91   3259    2711    5237
```

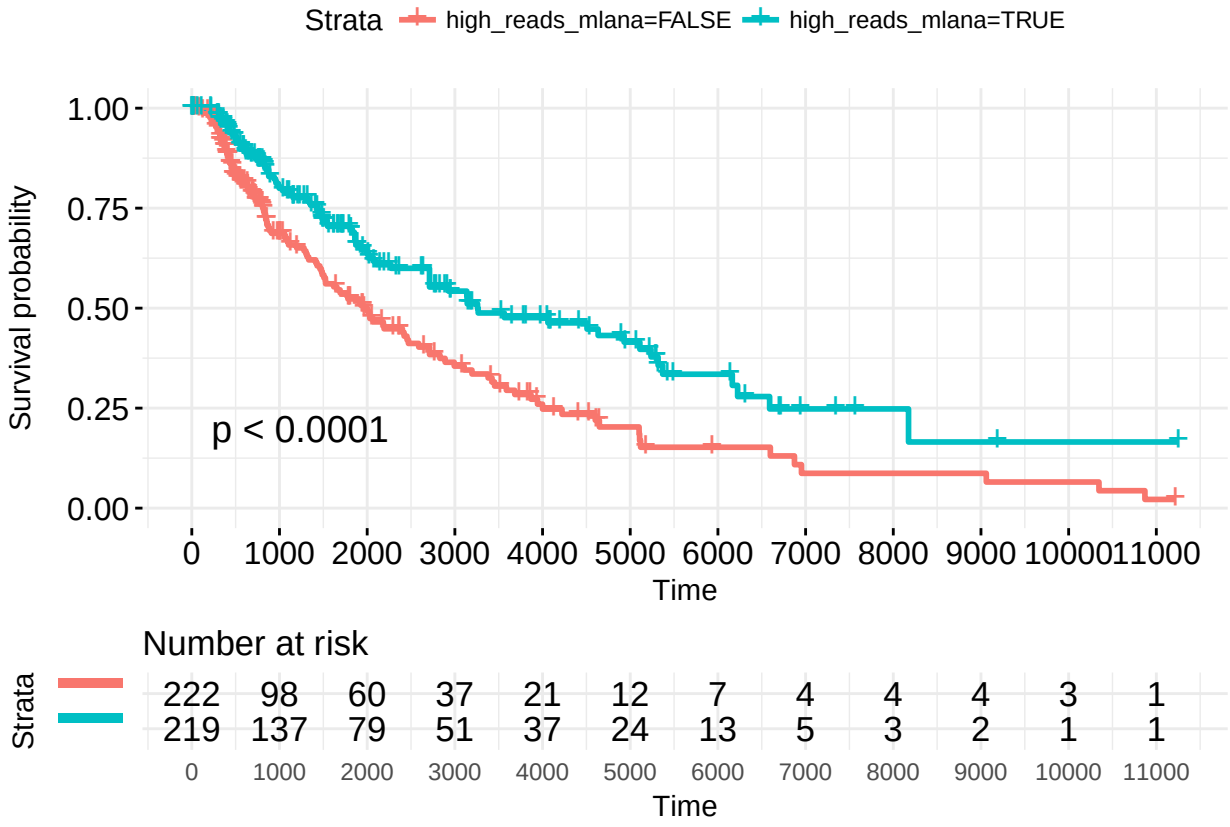
```
ggsurvplot(fit.1, data = df.annot, risk.table = T, pval = T, break.time.by = 1000,
           ggtheme = theme_minimal(), risk.table.y.text.col = T, risk.table.y.text = F)
```




```
fit.2 = survfit(Surv(time = survival, event = status == "dead") ~ high_reads_mlana,
                data = df.annot)
print(fit.2)
```

```
## Call: survfit(formula = Surv(time = survival, event = status == "dead") ~
##     high_reads_mlana, data = df.annot)
##
##    9 observations deleted due to missingness
##                           n events median 0.95LCL 0.95UCL
## high_reads_mlana=FALSE 222    119   1960    1524    2588
## high_reads_mlana=TRUE  220     90   3259    2711    5237
```

```
ggsurvplot(fit.2, data = df.annot, risk.table = T, pval = T, break.time.by = 1000,
           ggtheme = theme_minimal(), risk.table.y.text.col = T, risk.table.y.text = F)
```



Using ANOVA analysis and Cox regression, here number of reads is a continuous variable

```
# Here reads are reads coming from MLANA-specific clonotypes
# and reads_total are reads from all clonotypes

# Total reads is not significant given MLANA (P=0.31)

anova(coxph(formula = Surv(time = survival, event = status == "dead") ~ reads*reads_total,
            data = df.annot))
```

```
## Analysis of Deviance Table
##  Cox model: response is Surv(time = survival, event = status == "dead")
## Terms added sequentially (first to last)
##
##                    loglik   Chisq Df Pr(>|Chi|)
## NULL              -1064.3
## reads             -1059.2 10.2446  1   0.001371 **
## reads_total       -1058.7  1.0210  1   0.312290
## reads:reads_total -1054.6  8.1371  1   0.004337 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Adding total reads first -- still MLANA reads is near significant P=0.06

anova(coxph(formula = Surv(time = survival, event = status == "dead") ~ reads_total*reads,
            data = df.annot))
```

```
## Analysis of Deviance Table
##  Cox model: response is Surv(time = survival, event = status == "dead")
## Terms added sequentially (first to last)
```

```
##
##                     loglik  Chisq Df Pr(>|Chi|)
## NULL               -1064.3
## reads_total        -1060.4 7.7287  1   0.005435 **
## reads              -1058.7 3.5368  1   0.060021 .
## reads_total:reads -1054.6 8.1371  1   0.004337 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```