Code for generating plots R

Tobias Heyman

6 June 2021

Figure 3.2

This is the code that was used to generate Figure 3.2 of the master thesis.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(ggbeeswarm)
```

```
## Warning: package 'ggbeeswarm' was built under R version 4.0.4
```

```
library(ggbreak)
```

```
## Warning: package 'ggbreak' was built under R version 4.0.5
```

```
## ggbreak v0.0.9
##
## If you use ggbreak in published research, please cite the following
## paper:
##
## S Xu, M Chen, T Feng, L Zhan, L Zhou, G Yu. Use ggbreak to effectively
## utilize plotting space to deal with large datasets and outliers.
## Frontiers in Genetics. 2021, 12:774846. doi: 10.3389/fgene.2021.774846
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## 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(forcats)
```

```
## Warning: package 'forcats' was built under R version 4.0.3
```

```
# load the normal ranges of the tandem-repeats and load the observed tandem-repeat lengths from paitien

ranges <- read.delim("repeat_ranges_fig.txt")
data <- read.delim("detected_tandem_repeats_part1.txt")
```

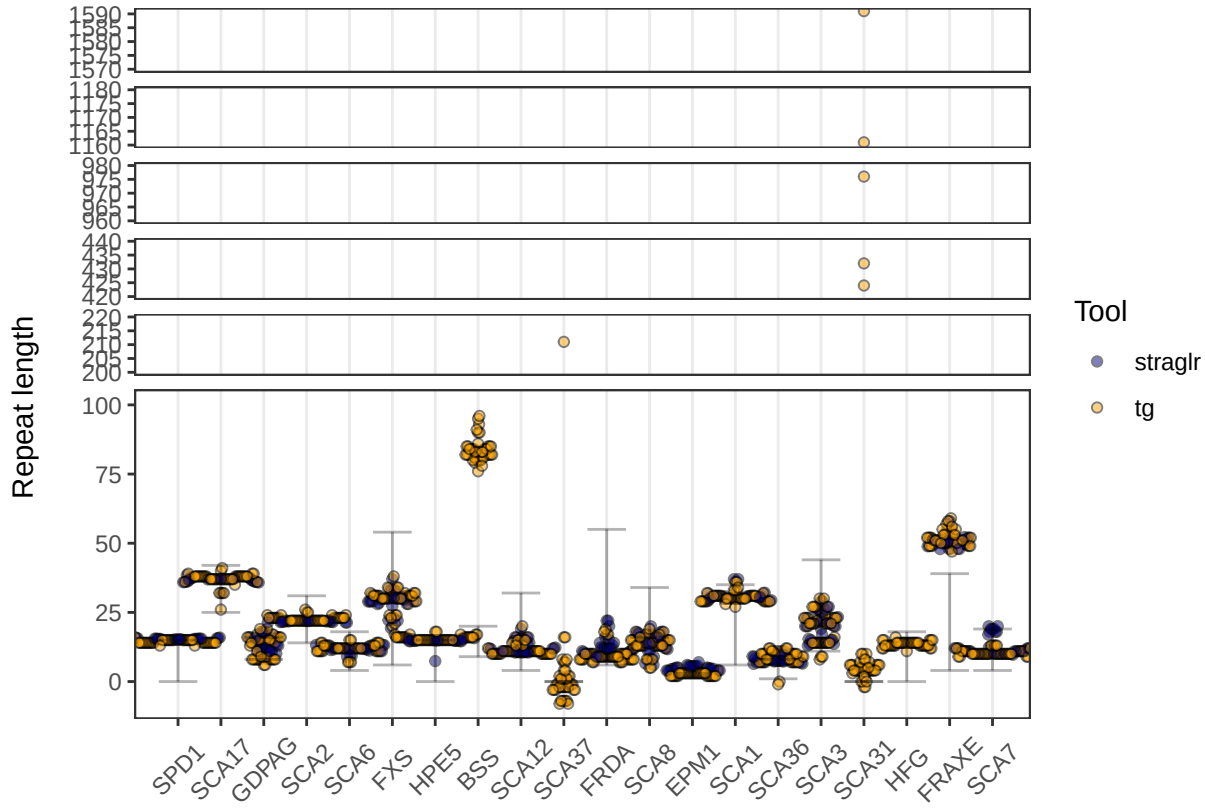
```r
data$length <- as.numeric(data$length)
```

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

```r
ranges$min <- as.numeric(ranges$min)
ranges$max <- as.numeric(ranges$max)

# order the diseases in the order that maximizes the vertical space between neighbouring diseases.

ordered <- data %>%
  mutate(disease = fct_relevel(disease,
            "SPD1", "SCA17","GDPAG","SCA2","SCA6","FXS","HPE5","BSS","SCA12","SCA37","FRDA","SCA8","EPM

ordered_ranges <- ranges %>%
  mutate(disease = fct_relevel(disease,
            "SPD1", "SCA17","GDPAG","SCA2","SCA6","FXS","HPE5","BSS","SCA12","SCA37","FRDA","SCA8","EPM


# only select normal ranges

ordered_ran <- ordered_ranges[ordered_ranges$type %in% c("normal"),]

ggplot() +
  geom_beeswarm(data=ordered, aes(y=length, x=disease, fill=Tool), cex = 0.17, priority= "density", alp
  geom_errorbar(data=ordered_ran, aes(min=min, max= max, x=disease), color="black", alpha=0.3)+
  theme_bw()+
  # change colors of the dots
  scale_fill_manual(values = c( "navyblue", "orange"))+
  xlab("Disease")+
  ylab("Repeat length")+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5), axis.title.x = element_blank(), panel.grid
  # add scale breaks
  scale_y_break(c(100, 200))+
  scale_y_break(c(440,960))+
  scale_y_break(c(980, 1160))+
  scale_y_break(c(1180, 1570))+
  scale_y_break(c(220, 420))
```

```
## Warning: Removed 85 rows containing missing values (position_beeswarm).
```

```
## Warning: Removed 85 rows containing missing values (position_beeswarm).
## Removed 85 rows containing missing values (position_beeswarm).
## Removed 85 rows containing missing values (position_beeswarm).
## Removed 85 rows containing missing values (position_beeswarm).
## Removed 85 rows containing missing values (position_beeswarm).
## Removed 85 rows containing missing values (position_beeswarm).
```



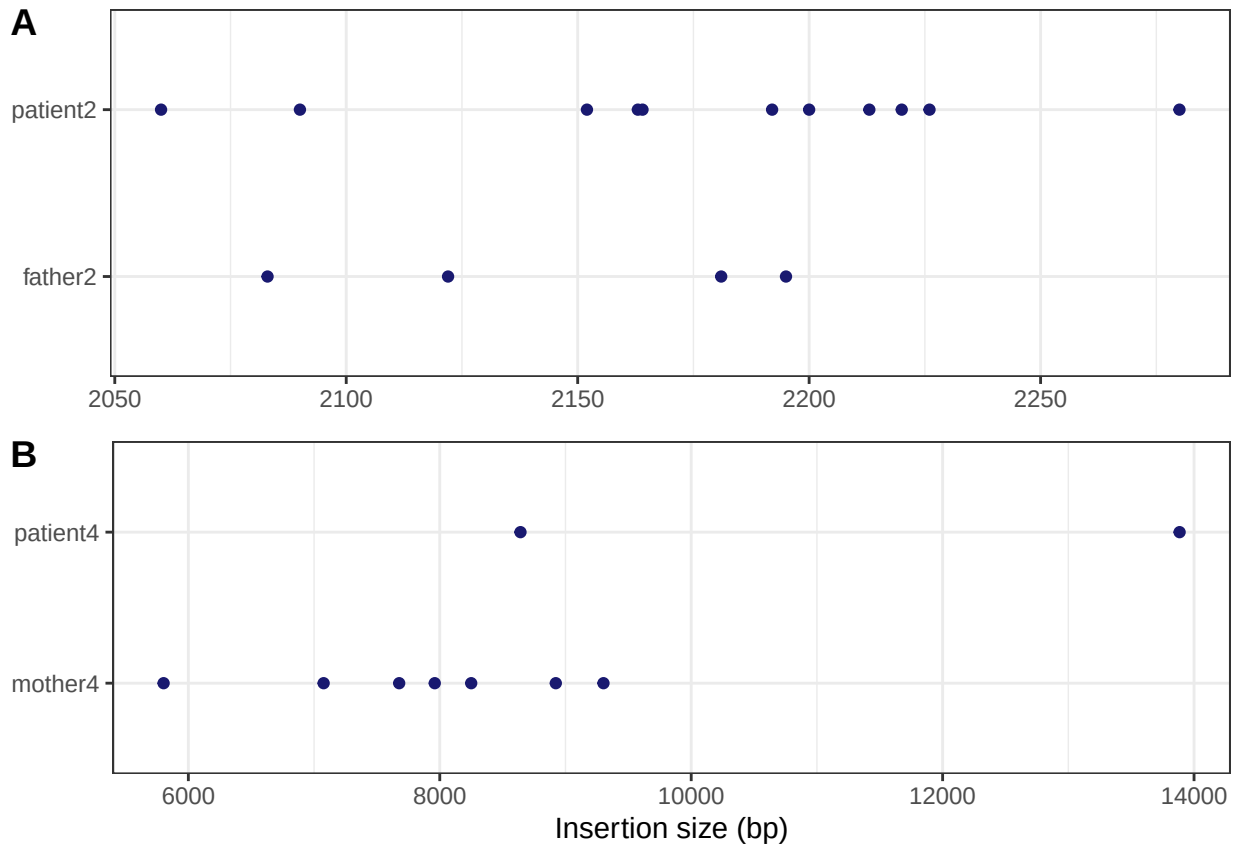

```
#ggsave("all_patients.png", width = 7, height = 8)
```

Figure 3.5

This is the code that was used to generate Figure 3.5 of the master thesis.

```
library(ggplot2)

# load the observed insertion sizes in the tandem-repeat associated with Spinocerebellar ataxia type 31

data <- read.delim("SCA_reads.txt")

# split the data into the two trios

dat2 <- data[data$individual %in% c("patient2", "father2"),]
dat4 <- data[data$individual %in% c("patient4", "mother4"),]

pat2 <- ggplot(data=dat2, aes(x = insertion.size, y= individual))+
  geom_point(color = "midnightblue")+
  theme_bw()+
  xlab("Insertion size (bp)")+
  theme(axis.title.y = element_blank(), axis.title.x = element_blank())

pat4 <- ggplot(data=dat4, aes(x = insertion.size, y= individual))+
  geom_point(color = "midnightblue")+
  theme_bw()+
```

```
  xlab("Insertion size (bp)")+
  theme(axis.title.y = element_blank())

# arrange the plots

cowplot::plot_grid(pat2, pat4, ncol = 1, labels = c("A", "B"))
```

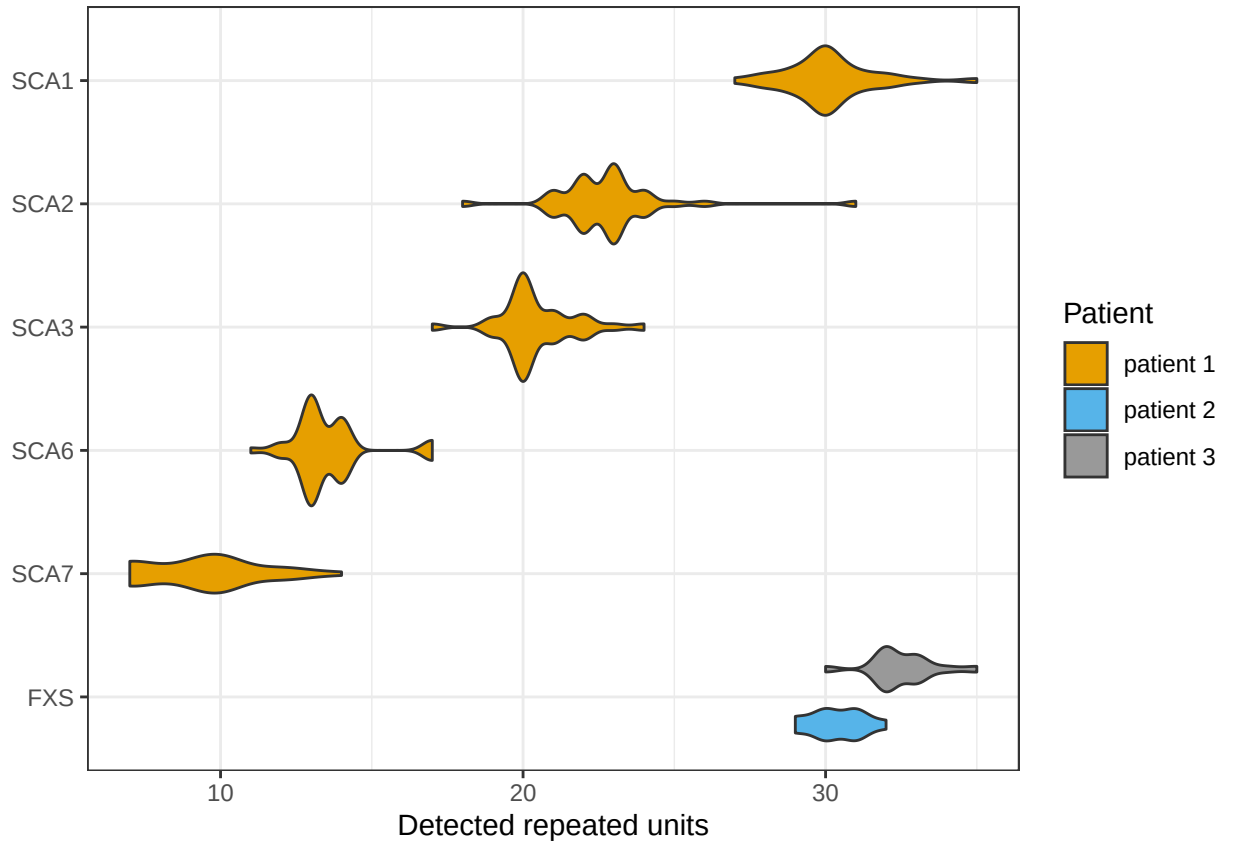



```
# ggsave("read_ranges.png", width = 8, height = 6)
```

Figure 3.6

This is the code that was used to generate Figure 3.6 of the master thesis.

```
library(ggplot2)


data <- read.delim("validation_reads.txt")

ggplot(data=data, aes(x = length, y= Disease, fill= Patient))+
  geom_violin()+
  theme_bw()+
  xlab("Detected repeated units")+
  scale_fill_manual(values = c("#E69F00", "#56B4E9", "#999999"))+
  theme(axis.title.y = element_blank())+
  scale_y_discrete(limits=c("FXS", "SCA7", "SCA6", "SCA3", "SCA2", "SCA1"))
```




```
# ggsave("validation_ranges.png", width = 8, height = 5)
```

Figure 3.8

This is the code that was used to generate Figure 3.8 of the master thesis.

```
library(ggplot2)
library(ggbeeswarm)
library(ggbreak)
library(dplyr)
library(forcats)

# load the normal ranges of the tandem-repeats and load the observed tandem-repeat lengths from paitien
# to telomere-to-telomere reference genome (T2T).

ranges <- read.delim("repeat_ranges_fig.txt")
data <- read.delim("detected_TR_T2T.txt")

data$length <- as.numeric(data$length)
ranges$min <- as.numeric(ranges$min)
ranges$max <- as.numeric(ranges$max)

# order the diseases in the same order as for figure 3.2

ordered <- data %>%
  mutate(disease = fct_relevel(disease,
```

```
            "SPD1", "SCA17","GDPAG","SCA2","SCA6","FXS","HPE5","BSS","SCA12","SCA37","FRDA","SCA8","EPM

ordered_ranges <- ranges %>%
  mutate(disease = fct_relevel(disease,
            "SPD1", "SCA17","GDPAG","SCA2","SCA6","FXS","HPE5","BSS","SCA12","SCA37","FRDA","SCA8","EPM

# group the normal and pathogenic ranges in separate data frames

ordered_ran <- ordered_ranges[ordered_ranges$type %in% c("normal"),]
ordered_pat <- ordered_ranges[ordered_ranges$type %in% c("pathogen") & (ordered_ranges$max < 120 ),]

# make plot T2T

plotT2T <- ggplot() +
  geom_beeswarm(data=ordered, aes(y=length, x=disease, color= Person, shape= Tool), cex = 1.05, priority
  geom_errorbar(data=ordered_ran, aes(min=min, max= max, x=disease), color="black", alpha=0.3)+
  geom_errorbar(data=ordered_pat, aes(min=min, max= max, x=disease), color="red")+
  theme_bw()+
  scale_color_manual(values = c( "navyblue", "grey", "orange"))+
  xlab("Disease")+
  ylab("Repeat length")+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5), legend.position = "none")+
  # change the scale of the y-axis to fixed length
  ylim(-7, 100)




# load the normal ranges of the tandem-repeats and load the observed tandem-repeat lengths from paitien
# to human reference genome 38 (hg38).

data <- read.delim("detected_tandem_repeats_part1.txt")

data$length <- as.numeric(data$length)
```

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

```
# order the diseases in the same manner as for figure 3.2

ordered_dat <- data %>%
  mutate(disease = fct_relevel(disease,
            "SPD1", "SCA17","GDPAG","SCA2","SCA6","FXS","HPE5","BSS","SCA12","SCA37","FRDA","SCA8","EPM

# only retain the information for trio 19.22059 (Patient 1)
ordered_dat <- ordered_dat[ordered_dat$Person %in% c("patient1", "mother1", "father1"),]

plothg38 <- ggplot() +
  geom_beeswarm(data=ordered_dat, aes(y=length, x=disease, color= Person, shape= Tool), cex = 1.05, pri
  geom_errorbar(data=ordered_ran, aes(min=min, max= max, x=disease), color="black", alpha=0.3)+
  geom_errorbar(data=ordered_pat, aes(min=min, max= max, x=disease), color="red")+
  theme_bw()+
  scale_color_manual(values = c( "navyblue", "grey", "orange"))+
  xlab("Disease")+
  ylab("Repeat length")+
```

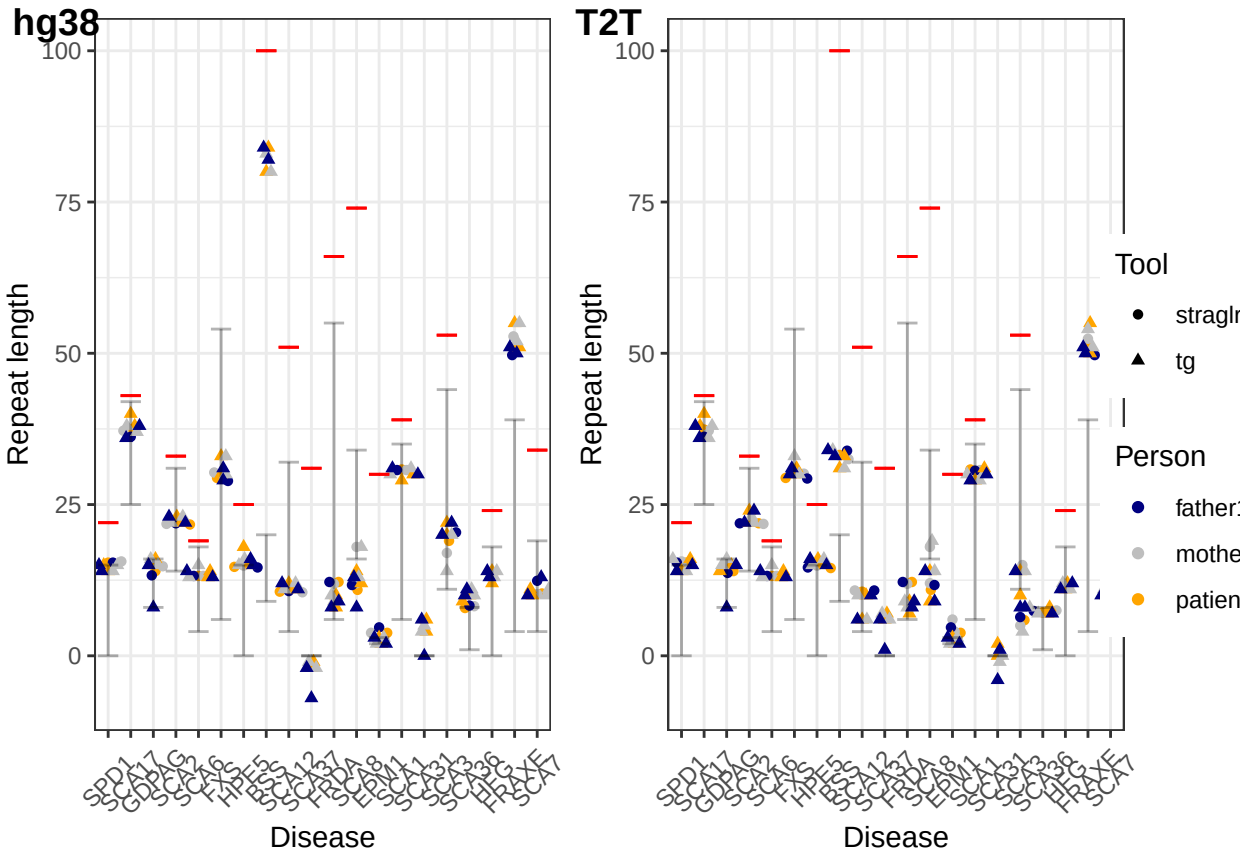
```
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5), legend.position = "none")+
  # change the scale of the y-axis to fixed length
  ylim(-7, 100)



# get the legend from plothg38 and plotT2T

legend <- cowplot::get_legend(ggplot() +
  geom_beeswarm(data=ordered_dat, aes(y=length, x=disease, color= Person, shape= Tool), cex = 1.15, pri
  geom_errorbar(data=ordered_ran, aes(min=min, max= max, x=disease), color="black", alpha=0.3)+
  geom_errorbar(data=ordered_pat, aes(min=min, max= max, x=disease), color="red")+
  theme_bw()+
  scale_color_manual(values = c( "navyblue", "grey", "orange")))
```

```
## Warning: Removed 12 rows containing missing values (position_beeswarm).
```

```
# arrange the two plots and their legend

cowplot::plot_grid(plothg38, plotT2T, legend, rel_widths = c(6,6,1.06), nrow = 1, labels = c("hg38", "T
```

```
## Warning: Removed 12 rows containing missing values (position_beeswarm).
```

```
## Warning: Removed 9 rows containing missing values (position_beeswarm).
```




```
#ggsave("patientT2T.png", width = 10, height = 5)
```

Figure 3.9

This is the code that was used to generate Figure 3.9 of the master thesis.

```
library(karyoploteR)
```

```
## Warning: package 'karyoploteR' was built under R version 4.0.3
## Loading required package: regioneR
## Warning: package 'regioneR' was built under R version 4.0.3
## Loading required package: GenomicRanges
## Warning: package 'GenomicRanges' was built under R version 4.0.3
## Loading required package: stats4
## Loading required package: BiocGenerics
## Warning: package 'BiocGenerics' was built under R version 4.0.5
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
## Loading required package: S4Vectors
## Warning: package 'S4Vectors' was built under R version 4.0.3
##
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:dplyr':
##
##     first, rename
## The following object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
## Warning: package 'IRanges' was built under R version 4.0.3
##
## Attaching package: 'IRanges'
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
## The following object is masked from 'package:grDevices':
##
##     windows
## Loading required package: GenomeInfoDb
## Warning: package 'GenomeInfoDb' was built under R version 4.0.5
```

```
library(ggplot2)

# load the genomic positions of tandem-repeats detected in a genome wide analysis by tandem-genotypes
RE_positions <- read.delim("QC/tandem-genotypes/proband_rmsk_pos.txt", header = T)

# load the genomic positions of tandem-repeats detected in a genome wide analysis by Straglr and TRiCoL
RE_positions_1 <- read.delim("QC/tandem-genotypes/proband_tristra_pos.txt", header = T)

# convert to the correct format
y <- toGRanges(RE_positions)

png("spread_rmsk_proband.png", width = 1500, height = 900)

kp <- plotKaryotype(genome = "hg38", chromosomes=paste0("chr", c(1:22, "X", "Y")))
kpPlotDensity(kp, data = y)

dev.off()
```

```
## pdf
##   2
```

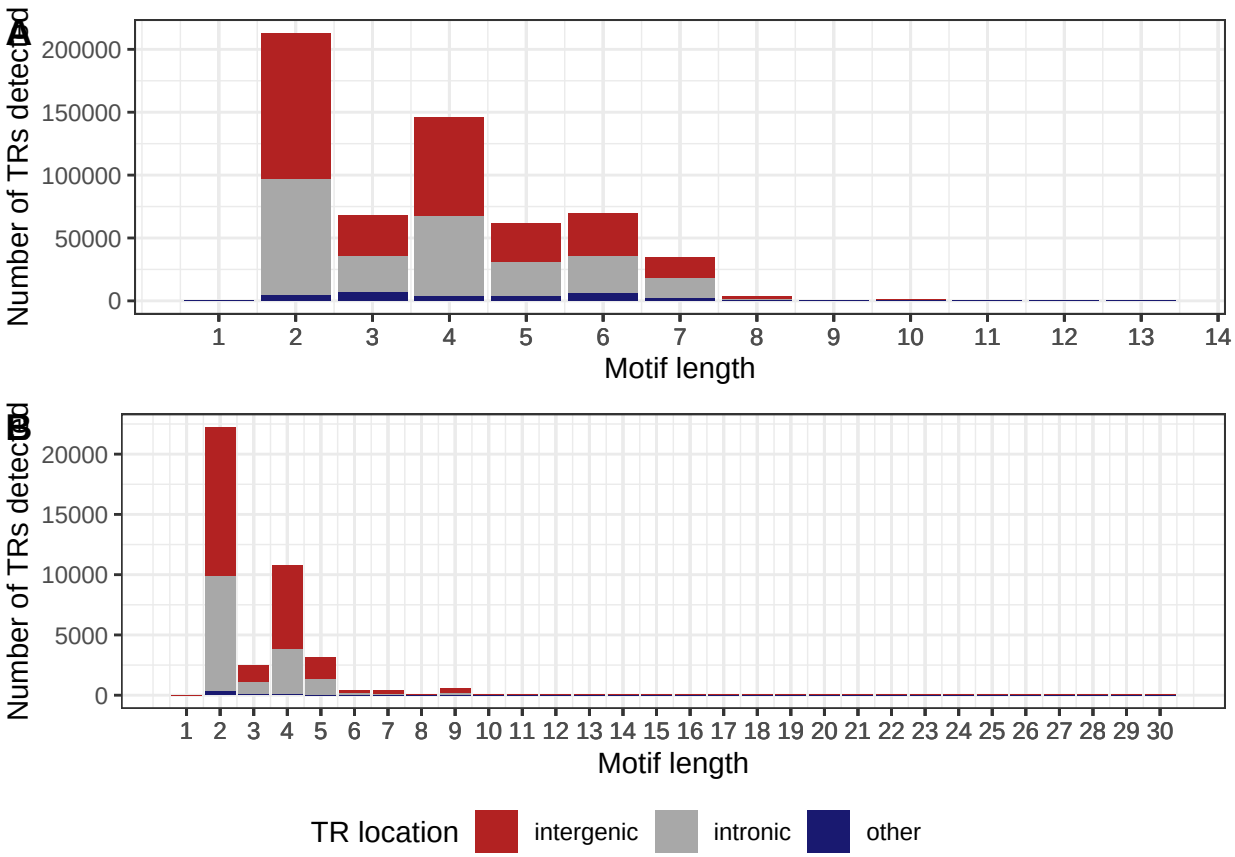
```
# make the same figure for Straglr and TRiCoLOR

# this row has a larger start position than end position
RE_positions_1 <- RE_positions_1[-c(41536),]

y1 <- toGRanges(RE_positions_1)

png("spread_TRiCoLOR_Stralr_proband.png", width = 1500, height = 900)

kp <- plotKaryotype(genome = "hg38", chromosomes=paste0("chr", c(1:22, "X", "Y")))
kpPlotDensity(kp, data = y1)

dev.off()
```

```
## pdf
##   2
```

Figure 3.10

This is the code that was used to generate Figure 3.10 of the master thesis.

```
library(ggplot2)

# load the data from the genome-wide analysis of tandem-genotypes

TR_data <- read.delim("QC/tandem-genotypes/prob_rmsk_k-bp.txt")


# load the data from the genome-wide analysis of Straglr and TRiCoLOR

TR_data_1 <- read.delim("QC/tandem-genotypes/prob_tristra_k-bp.txt")

plot_rmsk <- ggplot(data=TR_data, aes(x=motif, y=amount, fill=type)) +
  geom_bar(stat="identity")+
  theme_bw()+
  scale_fill_manual(values = c( "firebrick", "grey66", "midnightblue"))+
  ylab("Number of TRs detected")+
  xlab("Motif length")+
  scale_x_continuous(labels=as.character(TR_data_1$motif),breaks=TR_data_1$motif)+
  labs(fill="TR location")+
  theme(legend.position="none")

plot_tristra <- ggplot(data=TR_data_1, aes(x=motif, y=amount, fill=type)) +
  geom_bar(stat="identity")+
  theme_bw()+
  scale_fill_manual(values = c( "firebrick", "grey66", "midnightblue"))+
  ylab("Number of TRs detected")+
  xlab("Motif length")+
  scale_x_continuous(labels=as.character(TR_data_1$motif),breaks=TR_data_1$motif)+
  labs(fill="TR location") +
  theme(legend.position="none")

legend <- cowplot::get_legend(
  ggplot(data=TR_data_1, aes(x=motif, y=amount, fill=type)) +
  geom_bar(stat="identity")+
  theme_bw()+
  scale_fill_manual(values = c( "firebrick", "grey66", "midnightblue"))+
  labs(fill="TR location")
  )

# arrange the plots and the legend
ggpubr::ggarrange(plot_rmsk, plot_tristra, common.legend=T, labels = c("A", "B"), ncol = 1, legend = "b
```




```
#ggsave("TR_locations_proband_combo.png", width = 8, height = 8)
```

Supplementary table 1

```
library(dplyr)
library(forcats)

# read data
data <- read.delim("detected_tandem_repeats_part1.txt")

# only retain the first two columns
data <- data[,c(1,2)]
head(data)
```

```
##   disease length
## 1    SCA1   30.8
## 2    SCA1   30.7
## 3    SCA1   30.7
## 4    SCA1     30
## 5    SCA1     29
## 6    SCA1     30
```

```
# create new dataframe
differences <- setNames(data.frame(matrix(ncol = 3, nrow = 0)), c("disease", "sd", "mean"))
```

```r
diseases <- c("SPD1", "SCA17","GDPAG","SCA2","SCA6","FXS","HPE5","BSS","SCA12","SCA37","FRDA","SCA8","E

for (dis in diseases){
  # get the values in the right format
  vals <- unlist(data[data$disease == dis,], use.names = F)
  vals <- as.double(vals)
  vals <- na.omit(vals)
  vals <- as.double(vals)
  # calculate values
  sd <- round(sd(vals, na.rm = T), digits = 2)
  mean <- round(mean(vals, na.rm = T), digits = 2)
  entry <- data.frame(dis, sd, mean)
  names(entry) <- c("disease", "sd", "mean")
  # add the results to new dataframe
  differences <- rbind(differences, entry)

}
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

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

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

```
write.table(differences, "test.table.tsv", row.names = F, sep = "\t")
```