

Overview of svalignments package

Robert Scharpf

February 11, 2017

Contents

1	Introduction	1
2	Extract rearranged read pairs from a BAM file for a region of interest	1

1 Introduction

This package is used to extract properly paired reads and improperly paired reads from bam files. The main use-cases are (1) to identify rearranged read pairs that support structural variants, (2) assess BLAT realignment of selected reads for consistency with putative rearrangements and the whole-genome aligner. Most use-cases require a BAM file. For the puporse of illustrating examples in this vignette, we use BAM files in the svbams package.

2 Extract rearranged read pairs from a BAM file for a region of interest

This section describes how to extract rearranged read pairs from a BAM file for a specific genomic region of interest (e.g., a possible deletion). In the following code chunk, we load the svbams package which contains a small bam files for a region of chr15.

```
library(GenomicAlignments)
library(Rsamtools)
library(svbams)
library(svalignments)
path <- system.file("extdata", package="svbams")
```

First, we define the region of interest, loading the TxDb.Hsapiens.UCSC.hg19.refGene package to extract the sequence length for chromosome 15.

```
library(TxDb.Hsapiens.UCSC.hg19.refGene)
region <- GRanges("chr15", IRanges(63201003, 63209243))
si <- seqinfo(TxDb.Hsapiens.UCSC.hg19.refGene)
seqinfo(region) <- si["chr15", ]
```

To be sure this region is big enough to capture any interesting nearby features, we expand the region of interest to include 5kb on either side of the candidate sequence junctions.

```
region.expand <- GRanges("chr15", IRanges(start(region)-5e3, end(region)+5e3))
seqinfo(region.expand) <- seqinfo(region)
```

Next, we create a BamViews object that will contain metadata on the bamfiles.

```
bampath <- list.files(path, pattern="cgov44t.bam$", full.names=TRUE)
bview <- BamViews(bamPaths=bampath)
```

The following wrapper for ScanBamParams provides a default instance of the class with parameters for extracting improperly paired reads. Note that if which is missing for improperAlignmentParams, all improperly paired reads will be extracted from the BAM file. For large BAM files, this can be very slow.

```
iparams <- improperAlignmentParams(which=region.expand)
pparams <- properAlignmentParams(which=region.expand)
```

Next, we extract a GAlignmentsPairs object encapsulating all of the improperly and properly paired reads in this region.

```
irp <- getImproperAlignmentPairs(bview,
                                iparams,
                                mapq_thr=30,
                                use.mcols=TRUE)
prp <- getProperAlignmentPairs(bview,
                               pparams,
                               mapq_thr=30,
                               use.mcols=TRUE)
```

For visualization, we melt the GAlignmentPairs to GRanges and coerce to a data.frame. Depending on the size of the region, plotting all properly paired reads may not be desirable.

```
igr <- ga2gr(irp, is.improper=TRUE)
pgr <- ga2gr(prp, is.improper=FALSE)
pgr <- thinProperPairs(pgr, 10)
gr <- sortByRead1(c(igr, pgr))
df <- as(gr, "data.frame")
```

```
library(ggplot2)
library(scales)
colors <- c("#0072B2", "#009E73")
df$tagid <- as.numeric(df$tagid)
ggplot(df, aes(ymin=tagid-0.2, ymax=tagid+0.2,
               xmin=start/1e6, xmax=end/1e6, color=read,
               fill=read, group=tagid)) +
  geom_rect() +
  xlim(c(min(df$start), max(df$end))/1e6) +
  geom_line(aes(x=start/1e6, y=tagid)) +
  ylab("read pair index") +
  scale_x_continuous(breaks=pretty_breaks(5)) +
  geom_rect(data=as.data.frame(region),
            aes(xmin=start/1e6, xmax=end/1e6, ymin=-Inf, ymax=+Inf),
            fill="steelblue", color="transparent", alpha=0.2,
            inherit.aes=FALSE) +
  scale_color_manual(values=colors) +
  scale_fill_manual(values=colors) +
  xlab("Mb") +
  theme(axis.text.x=element_text(size=7),
        axis.text.y=element_blank()) +
  guides(fill=FALSE, color=FALSE)
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

