viewing alignments for structural rearrangements

Eitan Halper-Stromberg

January 29, 2013

Abstract

This package is designed to evaluate structural rearrangment calls from a candidate list, the output for tools such as HYDRA,GASV,VariationHunter, etc. The user should have a text file with one row per candidate structural rearrangment. For each candidate rearrangement, readpairs from the two loci will be read in and realigned three different ways. One of these realignments supports the structural variant, with readpairs realigned to a sequence representing the rearranged sequence (the sequence of the two loci concatenated together). The other two realignments support no structural rearrangement, with readpairs realigned to the two sequences representing contiguous fragments of the reference genome taken from each of the two loci.

1 Simple Usage

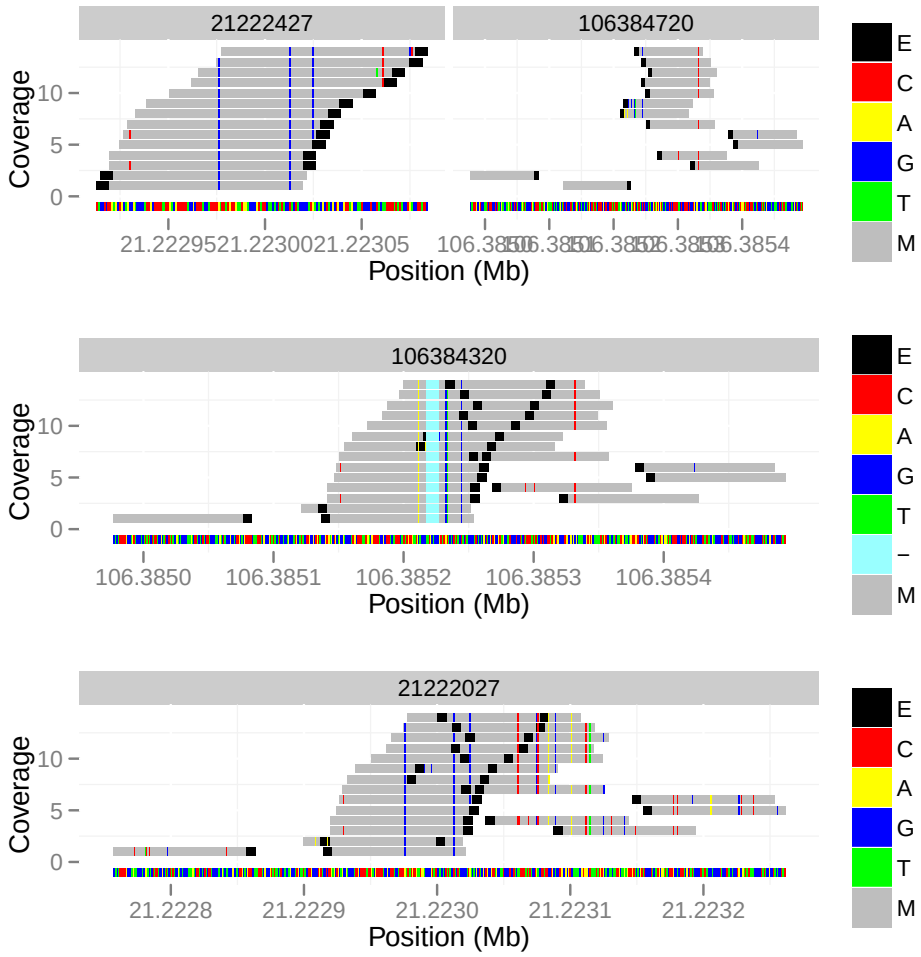
```
> library(targetSeqView)

> library(grid)
> library(doMC)
```

Perform realignment on 2 candidates that failed to validate

```
> path <- system.file("extdata", package="targetSeqView")
> nodes=1
> registerDoMC(nodes)
> filename=file.path(path,"twoSVJunctionsFailed.txt")
> retfail=ViewAndScore(filename=filename,bamFilePath=path,estimateIndelRate=FALSE,
+ estimateMmRate=FALSE,getReadLength=FALSE,build="hg19",verbose=TRUE)

[1] "Working on event 1 of 2"
[1] "primary alignment for event 1 done"
[1] "secondary alignment (1 of 2) for event 1 done"
[1] "secondary alignment (2 of 2) for event 1 done"
[1] "Working on event 2 of 2"
[1] "primary alignment for event 2 done"
[1] "secondary alignment (1 of 2) for event 2 done"
[1] "secondary alignment (2 of 2) for event 2 done"

> print('The likelihood scores for the events failing validation:')

[1] "The likelihood scores for the events failing validation:"

> print(retfail[[2]])

[1]  26.69819 132.74742
```

View the 3 realignment configurations for one of these failing events

```
>     p1=formatPlot(retfail[[1]][[1]][[1]][[2]],title='Alignment supporting a structural variant')
> p2=formatPlot(retfail[[1]][[1]][[2]][[2]],title='Alignment supporting no structural variant')
> p3=formatPlot(retfail[[1]][[1]][[3]][[2]],title='Alignment supporting no structural variant')
> grid.newpage()
> pushViewport(viewport(layout = grid.layout(3, 1)))
> print(p1,vp = viewport(layout.pos.row = 1, layout.pos.col=1))
> print(p2,vp = viewport(layout.pos.row = 2, layout.pos.col=1))
> print(p3,vp = viewport(layout.pos.row = 3, layout.pos.col=1))
```



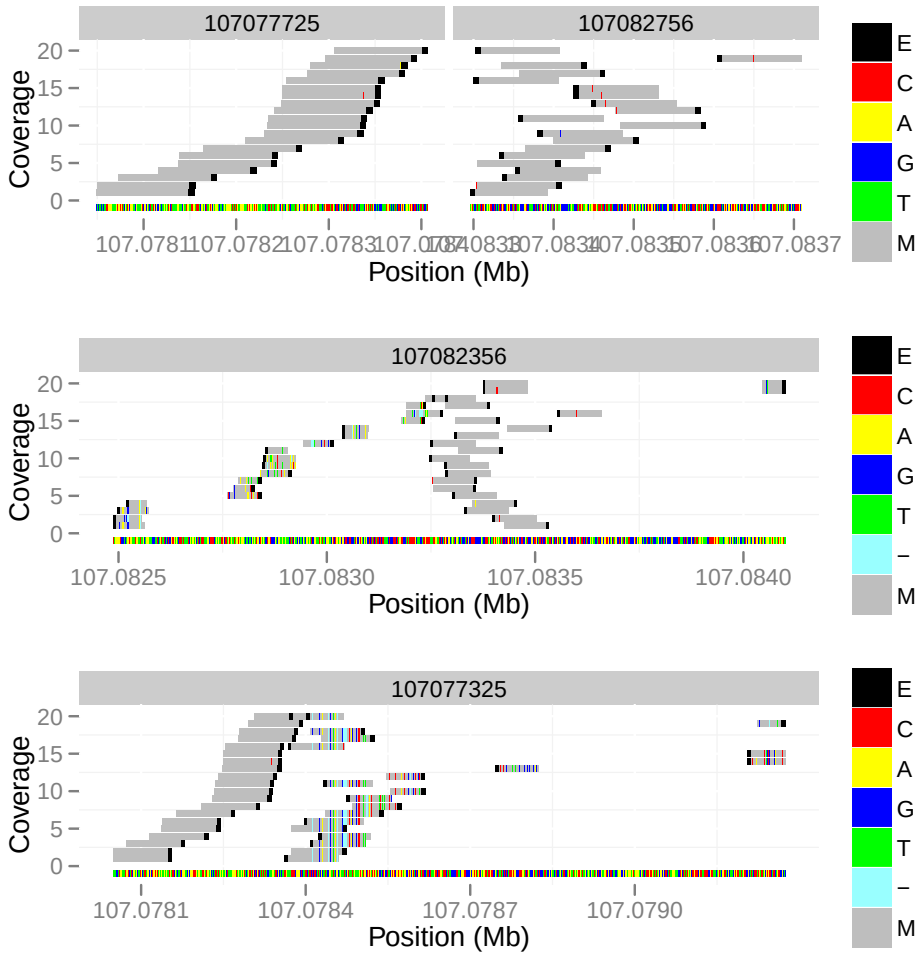
Perform realignment on 2 candidates that passed validation

```
> filename=file.path(path,"twoSVJunctionsPassed.txt")
> retpass=ViewAndScore(filename=filename,bamFilePath=path,estimateIndelRate=FALSE,
+ estimateMmRate=FALSE,getReadLength=FALSE,build="hg19",verbose=TRUE)

[1] "Working on event 1 of 2"
[1] "primary alignment for event 1 done"
[1] "secondary alignment (1 of 2) for event 1 done"
[1] "secondary alignment (2 of 2) for event 1 done"
[1] "Working on event 2 of 2"
[1] "primary alignment for event 2 done"
```

```
[1] "secondary alignment (1 of 2) for event 2 done"
[1] "secondary alignment (2 of 2) for event 2 done"

> print('The likelihood scores for the events passing validation:')

[1] "The likelihood scores for the events passing validation:"

> print(retpass[[2]])

[1] 2248.1051  463.7081
```

View the 3 realignment configurations for one of these passing events

```
> p1=formatPlot(retpass[[1]][[1]][[1]][[2]],title='Alignment supporting a structural variant')
> p2=formatPlot(retpass[[1]][[1]][[2]][[2]],title='Alignment supporting no structural variant')
> p3=formatPlot(retpass[[1]][[1]][[3]][[2]],title='Alignment supporting no structural variant')
> grid.newpage()
> pushViewport(viewport(layout = grid.layout(3, 1)))
> print(p1,vp = viewport(layout.pos.row = 1, layout.pos.col=1))
> print(p2,vp = viewport(layout.pos.row = 2, layout.pos.col=1))
> print(p3,vp = viewport(layout.pos.row = 3, layout.pos.col=1))
```




```
> toLatex(sessionInfo())
```

- R version 2.15.2 Patched (2012-10-28 r61038), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.iso885915`, `LC_NUMERIC=C`, `LC_TIME=en_US.iso885915`, `LC_COLLATE=C`, `LC_MONETARY=en_US.iso885915`, `LC_MESSAGES=en_US.iso885915`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.iso885915`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, utils
- Other packages: BSgenome 1.26.1, BSgenome.Hsapiens.UCSC.hg19 1.3.19, BiocGenerics 0.4.0, Biostrings 2.26.2, GenomicRanges 1.10.5, IRanges 1.16.4, Rsamtools 1.10.2, doMC 1.2.5, foreach 1.4.0, ggplot2 0.9.0, iterators 1.0.6, multicore 0.1-7, targetSeqView 1.0
- Loaded via a namespace (and not attached): MASS 7.3-22, RColorBrewer 1.0-5, bitops 1.0-5, codetools 0.2-8, colorspace 1.2-0, compiler 2.15.2, dichromat 1.2-4, digest 0.6.0, labeling 0.1, memoise 0.1, munsell 0.4, parallel 2.15.2, plyr 1.7.1, proto 0.3-9.2, reshape2 1.2.1, scales 0.2.2, stats4 2.15.2, stringr 0.6.1, tools 2.15.2, zlibbioc 1.4.0