Finding Structural Variants in Short Read, Paired-end Sequence Data with R and Bioconductor

Sean Davis
National Cancer Institute,
National Institutes of Health
Bethesda, MD, USA
sdavis2@mail.nih.gov

July 2, 2010

Abstract
Second-generation sequencing technologies, when used to sequence genomic DNA with paired-end reads, are extremely powerful and sensitive tools for discovering structural variation within a genome. Insertions and deletions as well as translocations (and inversions—not discussed directly here), if present in the genomic DNA under investigation, may be captured by examining paired-end reads that are “unusual”. This little vignette lays out the concepts for finding these structural variants in a simulated genome (taken from human chromosome 17).

1 Introduction
In both “normal” and “disease” states, the structure of the genome is a potentially important factor in determining biology. Insertions of DNA can alter the regulatory regions or coding sequence of genes by introducing foreign regulatory elements or by disrupting coding sequence or splice sites. Deletions can have similar effects by removing functional elements in DNA. Translocations can place one functional element (such as a transcription-factor binding site) into proximity of another functional element such as a miRNA or proteing-coding gene, thus affecting the normal regulatory pathways. Furthermore, translocations may act more directly by producing fusion transcripts or chimeras of two protein-coding genes; this particular phenomenon has been described in numerous types of cancer.
1.1 Paired-end reads

Paired-end reads can encode information about the presence and nature of structural variants in the genome. Medvedev et al. provide an excellent review in a recent manuscript ([4]). Paired-end reads are generated in a relatively straightforward way. As review, the paired-end sequencing process at a very high level looks like:

1. Genomic DNA is sheared into fragments
2. The sheared fragments of DNA are size-selected (the target size is based on the particular sequencing technology), manipulated to make them suitable for sequencing (library preparation), and finally placed on the sequencer
3. One end of the pair (the primary read) is read up to the read length (often between 30 and 100 bp, for Illumina)
4. The other end of the pair (the secondary read) is read, again up to the read length
5. The paired ends are aligned to the reference genome resulting in a map of where the ends of the original genomic fragments originated

Note that in point 2 above, the fragments are size-selected. Therefore, if all the DNA that was finally prepared for paired-end sequencing was without structural variation, the distances between the two paired-ends—the insert size—should recapitulate the distribution of fragment sizes that were selected. If a pair exhibits an insert size that is an extreme outlier—an outlier pair—it could be evidence for a structural variation.

1.2 Insert size and structural variation

The insert size should be systematically affected by structural variation (Table 1, Figure 1).

Table 1: Effects of structural variants on apparent insert size (the insert size as measured by alignment back to the reference genome).

<table>
<thead>
<tr>
<th>Structural Variant</th>
<th>Affect on Apparent Insert Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small insertion</td>
<td>Decrease</td>
</tr>
<tr>
<td>Small deletion</td>
<td>Increase</td>
</tr>
<tr>
<td>Large deletion</td>
<td>Increase</td>
</tr>
<tr>
<td>Intrachromosomal translocation</td>
<td>Increase</td>
</tr>
<tr>
<td>Interchromosomal translocation</td>
<td>Change mate chromosome</td>
</tr>
<tr>
<td>Large insertion</td>
<td>Increase or lack of mapped mate</td>
</tr>
</tbody>
</table>
A tandem duplication will create an everted duplication linking signature, with the inserted sequence (green) is copied from another location in the genome. (f) A linked insertion signature is composed of two linking signatures and arises when the inserted sequence (green) is copied from another location in the genome. (g,h) In the anchored split mapping signature, one mate has a good mapping, whereas the other has a split mapping. For a deletion (g) the prefix and suffix surround the deletion, whereas for an insertion (h) the split read has the prefix and suffix mapped to adjacent locations, while a middle part does not map. (i) When a novel genomic segment is inserted, a hanging insertion signature is created, in which only one of the mates has a good mapping. From [4], figure 1.

Figure 1: Illustrations of PEM signatures. Mate pairs are sampled from the donor, where they are ordered with opposite orientation (the blue mate follows the orange), and are mapped to the reference (ref). Basic signatures include (a) insertions and (b) deletions, where the mapped distance is different from the insert size, as well as (c) inversions, where the order of the two mates is preserved but one of them changes orientation. (d) The linking signature has several discordant mate pairs with similar mapped distances identifying adjacency in the donor (dashed orange arrows) of two distal segments of the reference. The orientation and order of the mapped mate pairs depends on the orientation and order of the two segments in the reference; here, these are unchanged. (e) A linked insertion signature is composed of two linking signatures and arises when the inserted sequence (green) is copied from another location in the genome. (f)
2 Finding some structural variants

2.1 Data description

In order to keep the exercises manageable and to have a “gold standard” against which to check the results, I have generated an “abnormal” segment of human chromosome 17 that includes several structural variants (see Table 2). The code to generate the artificial sequence is given in this vignette. The MAQ package ([2]) was used to simulate 100000 pairs of reads (so 200000 reads) with insert sizes with mean of 200 and a standard deviation of 20. Some errors were allowed (error rates modeled after typical 36-basepair reads from Illumina). The reads were aligned to the human genome using the BWA software package ([3]) with default settings. The samtools package ([1]) was used to convert the output from BWA into BAM format.

Table 2: Description of insertions and deletions included in the simulated sequence. In addition, a segment of chromosome 17 between 40400000 and 40500000 was “translocated” from its original location to be attached between 40040000 and 40070000. Finally, five copies of the region between 40100000 and 40110000 were attached tandemly to create a copy number gain region.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Start</th>
<th>End</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>deletion</td>
<td>40020000</td>
<td>40020020</td>
<td>20</td>
</tr>
<tr>
<td>deletion</td>
<td>40022000</td>
<td>40022040</td>
<td>40</td>
</tr>
<tr>
<td>deletion</td>
<td>40024000</td>
<td>40024060</td>
<td>60</td>
</tr>
<tr>
<td>deletion</td>
<td>40026000</td>
<td>40026080</td>
<td>80</td>
</tr>
<tr>
<td>deletion</td>
<td>40028000</td>
<td>40028100</td>
<td>100</td>
</tr>
<tr>
<td>deletion</td>
<td>40030000</td>
<td>40030200</td>
<td>200</td>
</tr>
<tr>
<td>deletion</td>
<td>40032000</td>
<td>40034000</td>
<td>2000</td>
</tr>
<tr>
<td>insertion</td>
<td>40600000</td>
<td>NA</td>
<td>20</td>
</tr>
<tr>
<td>insertion</td>
<td>40602000</td>
<td>NA</td>
<td>100</td>
</tr>
<tr>
<td>insertion</td>
<td>40604000</td>
<td>NA</td>
<td>1000</td>
</tr>
</tbody>
</table>

```r
> library(BSgenome.Hsapiens.UCSC.hg19)
> ## start with 1MB of chr17
> a <- getSeq(Hsapiens, ‘chr17’, start=40000000, width=1000000)
> ## Generate a little random sequence for insertion
> b <- paste(sample(DNA_BASES, 10000, replace=TRUE), collapse=””)
> ## Now, paste together subsets of the sequences to
> ## generate the abnormal chromosome 17 segment
> newseq <- paste(substr(a, 1, 20000),
>                  # start deletions
>                  substr(a, 20020, 22000),
>                  substr(a, 22040, 24000),
>                  b,
>                  substr(a, 40020000, 40020020),
>                  substr(a, 40022000, 40022040),
>                  substr(a, 40024000, 40024060),
>                  substr(a, 40026000, 40026080),
>                  substr(a, 40028000, 40028100),
>                  substr(a, 40030000, 40030200),
>                  substr(a, 40032000, 40034000),
>                  substr(a, 40600000, 40602000),
>                  substr(a, 40604000, 40604000),
>                  substr(a, 40040000, 40070000),
>                  substr(a, 40100000, 40110000))
```
> f <- file("/tmp/sample.sequence.fa","w")
> writeLines(sample, f)
> writeLines(newseq, f)
> close(f)

### 2.2 Data import and preprocessing

Since the data are in BAM format, which is a binary version of the SAM format, the Rsamtools package will be used for reading in the data. The SAM format specification provides for storage of paired-end reads. Both ends of the pair are stored, regardless of whether or not they are mapped. In addition, each read record has information about the mapping of its mate. Since we are typically going to be interested in operating on intervals with data attached to each interval, a natural choice of data structure is the GRanges class. A little convenience function, scanBamToGRanges, does the necessary reading and minor preprocessing for our BAM file of reads.

> library(GenomicRanges)
> library(Rsamtools)
> scanBamToGRanges <- function(...) {
+   dat <- scanBam(...)[[1]]
+ }
```
+ keep <- !is.na(dat$pos)
+ GRanges(seqnames=dat$rname[keep],
+    ranges=IRanges(start=dat$pos[keep], width=nchar(dat$seq[keep])),
+    strand=dat$strand[keep], isize=dat$isize[keep],
+    mrnm=dat$mrnm[keep], flag=dat$flag[keep])
+ }

> fname <- system.file('extdata', 'reads.sorted.bam', package='EMBL2010')
> totalGR <- scanBamToGRanges(fname)
> length(totalGR)
[1] 199403
> head(totalGR, 3)

GRanges with 3 ranges and 3 elementMetadata values

```

```
seqnames ranges strand | isize mrnm
<Rle> <IRanges> <Rle> | <integer> <factor>
[1] chr10 [ 894815, 894849] + | 0 chr14
[2] chr10 [4256528, 4256562] - | 0 chr17
[3] chr10 [5954230, 5954264] - | 0 chr17
flag
<integer>
[1] 161
[2] 81
[3] 113

seqlengths
chr10 chr11 chr12 chr13 chr14 ... chr9 chrM chrX chrY
NA NA NA NA NA ... NA NA NA NA

The first point to note is that, while all the sequence in the sample sequence file originated from the chr17 human reference sequence, there are mappings to other chromosomes. Since the simulation allows for errors, such errors can induce enough entropy into the original sequence to facilitate alignment to another region of the genome. The effect is small, but non-trivial.

```
> table(seqnames(totalGR))

```

```
chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17
157 118 178 104 129 123 140 196093
chr18 chr19 chr1 chr20 chr21 chr22 chr2 chr3
68 151 330 80 35 65 236 211
chr4 chr5 chr6 chr7 chr8 chr9 chrX chrY
174 193 156 222 139 132 146 23

Using the information contained in the “flag” column of the elementMetadata, it is possible to get an overview of the mapping characteristics of the pairs; for example, which reads have an unmapped mate?

```
> library(bitops)
> # find those reads with unmapped mates
> # The flag for an unmapped mate is 0x0008
> summary(bitAnd(values(totalGR)$flag, 0x0008) == 0x0008)

       Mode   FALSE   TRUE  NA's
logical 199012    391     0

These reads with unmapped mates may (among other possibilities) represent insertions of foreign sequence relative to the reference.

2.3 Clustering outlier read pairs

As discussed above, the insert size is expected to change in systematic ways when structural variants are present. Those read pairs that show a larger or smaller insert size or have an unmapped mate represent “outlier read pairs”. For small insertions, there may be bridging reads that show a smaller-than-expected insert size. To test the feasibility of this thinking, I first restrict the GRanges object totalGR to come from chr17.

> chr17GR <- totalGR[seqnames(totalGR) == 'chr17']

Using the robust estimates of the mean and standard deviation, the median and median absolute deviation (mad), it is possible to choose a range of “normal” insert sizes. Of course, the isize column of the totalGR contains this information, but it only makes sense to compute on those reads that have a mapped mate and are on the same chromosome.

> median(abs(values(totalGR)$isize), na.rm=TRUE)

[1] 200

> mad(abs(values(totalGR)$isize), na.rm=TRUE)

[1] 20.7564

Note that these agree quite well with the parameters from the simulation. So, filtering the reads to those with isizes smaller than 137 (approximately $\mu - 3\sigma$), we can find potential pairs associated with small insertions. More specifically, if there is clustering of reads that show smaller-than-expected insert sizes, there may be evidence for a deletion in the vicinity of the clustering. To calculate clustering, it is fairly straightforward to use ideas similar to those demonstrated for ChIP-seq analysis.

To look for the reads that have a mapped mate that also maps to chr17 and have insert sizes less than 137 bases long, we use the following incantation. Basically, the first command finds reads that have a mapped mate where the mate is mapped to chromosome 17 and the insert size is smaller than expected. The reads are then extended to half of the expected insert size. Coverage is calculated...
on the extended reads and finally the coverage vector is sliced (thresholded) to
find regions that are somewhat enriched in short-insert-size pairs. The last step
would require some statistical modeling to determine the best threshold; here,
a convenient threshold was chosen empirically.

```r
> smallInsertGR <- chr17GR[(bitAnd(values(chr17GR)$flag, 0x0008) != 0x0008) &
+ (values(chr17GR)$mrnm == 'chr17') & (abs(values(chr17GR)$isize) < 137)]
> smallInsertGR <- resize(smallInsertGR, 135)
> cvgSI <- coverage(smallInsertGR)['chr17']
> sliceSI <- slice(cvgSI, 10)
```

Reads with insert sizes larger than expected (263 bp, approximately $\mu + 3\sigma$)
may be associated with regions of the genome that are physically attached but
on the reference sequence are actually far apart. Deletions and translocations
would be expected to show reads that have larger-than-expected insert sizes.
Again, we find these reads and then cluster them to find putative regions of
deletion or translocation. Using an analogous set of functions to short-insert-
size pairs, it is possible to cluster the long-insert-size pairs.

```r
> largeInsertGR <- chr17GR[(bitAnd(values(chr17GR)$flag, 0x0008) != 0x0008) &
+ (values(chr17GR)$mrnm == 'chr17') & (abs(values(chr17GR)$isize) > 263)]
> largeInsertGR <- resize(largeInsertGR, 135)
> cvgLI <- coverage(largeInsertGR)['chr17']
> sliceLI <- slice(cvgLI, 4)
```

Finally, it is possible to do the same with reads with unmapped mates.

```r
> unmappedGR <- chr17GR[(bitAnd(values(chr17GR)$flag, 0x0008) == 0x0008)]
> unmappedGR <- resize(unmappedGR, 135)
> cvgUM <- coverage(unmappedGR)['chr17']
> sliceUM <- slice(cvgUM, 5)
```

### 2.4 Interpretation of results

Currently, there is little infrastructure in place to facilitate the interpretation of
the results of clustering outlier read pairs. Using the large-insert-pairs clustering
result as an example is still instructive, though, for gaining insight into the
structural variation present. Examining the text output of the sliced coverage
is informative, but leaves a sour feeling in the lack of insight it provides.

```r
> sliceLI[['chr17']]
```

Views on a 78092744-length Rle subject

views:

<table>
<thead>
<tr>
<th>start</th>
<th>end</th>
<th>width</th>
</tr>
</thead>
<tbody>
<tr>
<td>40021994</td>
<td>40021997</td>
<td>4 [4 4 4 4]</td>
</tr>
</tbody>
</table>
However, some simple plots can be very informative about the fine- and large-scale structure of this artificial segment of chromosome 17. A little plotting function, `plotSV`, generates a plot showing the regions captured by the clustering process and then overlays the connections as encoded in the paired-end reads that overlap with the clusters. The resulting plots can be instructive (indeed, they led me to a bug in the generation of the original sequence).

```r
plotSV <- function(x, cvgslice, seqnm='chr17', ...) {
  xl <- min(seqselect(start(x), seqnames(x) == seqnm))
  xr <- max(seqselect(end(x), seqnames(x) == seqnm))
  par(las=2)
  plot(c(xl,xr), c(0,3), pch='!', xlab="genomic coordinates",
       ylab='', axes=FALSE, ...)
  for(i in 1:length(cvgslice[[seqnm]])) {
    rect(start(cvgslice[[seqnm]])[i], 0, end(cvgslice[[seqnm]])[i], 0.5, 
         col='red', border='red')
  }
  subgr <- 
  subsetByOverlaps(x,
         GRanges(seqnames=seqnm,
         ranges=IRanges(start=start(cvgslice[[seqnm]]),
         end=end(cvgslice[[seqnm]]))))
  for(i in seq_len(length(subgr))) {
    strandq <- ifelse(bitAnd(values(subgr)$flag[i], 0x0010) == 0x0010, '-', '+')
    strandmate <- ifelse(bitAnd(values(subgr)$flag[i], 0x0020) == 0x0020, '-', '+')
    if(strandq == '+' & strandmate == '-') col <- 'red'
    else if(strandq == '-' & strandmate == '+') col <- 'green'
    else if(strandq == '+' & strandmate == '-') col <- 'blue'
    else if(strandq == '-' & strandmate == '+') col <- 'orange'
  }
```
Using this function, it is possible to generate some interesting plots. Figures 2 and 3 show the entire 1MB region and a zoomed-in view of the region with a number of small deletions, respectively.

```r
> pdf('fullRegionLI.pdf', width=9, height=7)
> plotSV(chr17GR, sliceLI, xlim=c(4e7, 4.1e7))
> dev.off()

> pdf('zoomDeletionsLI.pdf', width=9, height=7)
> plotSV(chr17GR, sliceLI, xlim=c(4.002e7, 4.0045e7))
> dev.off()
```

3 Conclusions and future work

While multiple software packages for finding structural variants exist (see table 1 in [4]), the R and Bioconductor tools offer an opportunity to build on existing infrastructure for short-read data analysis, to experiment with methodologies for extending existing methods, and to capitalize on the extremely powerful graphics and data handling capabilities inherent in R. There is a need to improve handling of paired-end reads (and related mate-pair reads) in a memory- and computationally-efficient manner. Also, clustering of reads could be done using other models or statistical or computational techniques. After the processing of reads down to clusters is complete, a mechanism to help to truly understand the variants that lead to those clustering results, such as a graph-theoretic approach to linking clusters, needs to be developed. Finally, and perhaps most difficult, is to come to some new biologic understanding or knowledge (are fusion genes formed, regulatory regions disrupted, etc.) based on the results.

4 sessionInfo()

```r
> toLatex(sessionInfo())
```
Figure 2: A view of the entire 1MB region showing some large-scale rearrangements. The red bars represent the locations of the clusters of abnormal read pairs. The lines demonstrate how the reads that map to the locations covered by the clusters connect to their mates. The colors of the lines correspond to the combinations of strand information in the two ends of the pair; read means the first pair is + and the second is - strand while green is the opposite. All the lines are drawn from the first pair location on the lower part of the plot to the second pair of the read on the upper part of the plot. A translocation is clearly visible as is the region of copy-number gain (which is a
Figure 3: A zoomed-in view of some small deletions. The markup of the figure is as for Figure 2. Refer to Table 2 for details of the locations of the deletions. Note that there is what appears to be a “extra” large deletion–this was a mistake in the code for generating the artificial sequence that I caught only after making the plots.
References


