Exploring the Ranges Infrastructure

Michael Lawrence

July 21, 2017

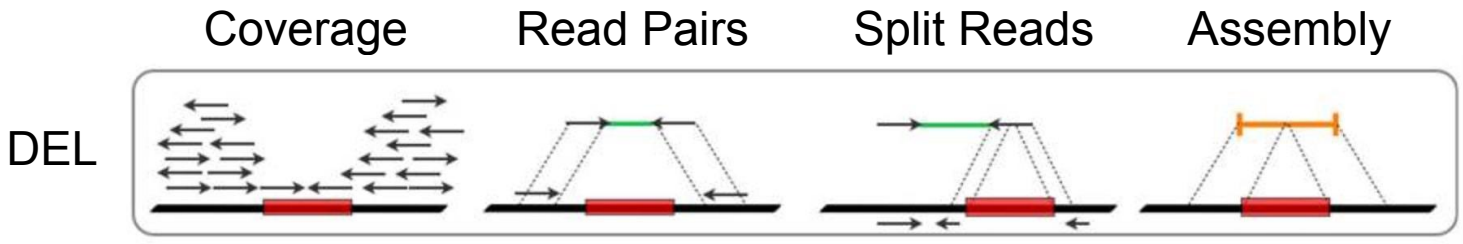
Outline

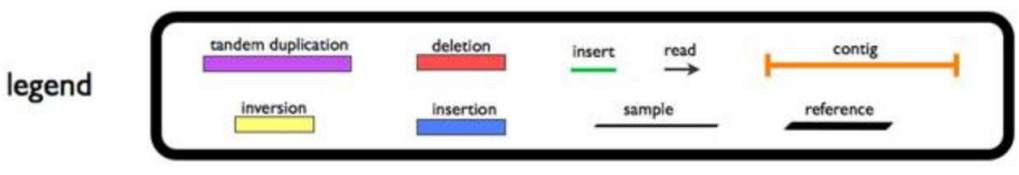
Example workflow: Structural variants

Structural variants are important for disease

- SVs are rarer than SNVs
  - SNVs: ~ 4,000,000 per genome
  - SVs: 5,000 - 10,000 per genome
- However, SVs are much larger (typically $>$ 1kb) and cover more genomic space than SNVs.
- The effect size of SV associations with disease is larger than those of SNVs.
  - SVs account for 13% of GTEx eQTLs
  - SVs are 26 - 54 X more likely to modulate expression than SNVs (or indels)

Detection of deletions from WGS data





Goal

Evaluate the performance of lumpy, a structural variant caller

Data

- Simulated a FASTQ containing known deletions using varsim
- Aligned the reads with BWA
- Ran lumpy on the alignments

Overview

1. Import the lumpy calls and truth set
2. Tidy the data
3. Match the calls to the truth
4. Compute error rates
5. Diagnose errors

Data import

Read from VCF:

```
library(RangesTutorial2017)
calls <- readVcf(system.file("extdata", "lumpy.vcf.gz",
                             package="RangesTutorial2017"))
truth <- readVcf(system.file("extdata", "truth.vcf.bgz",
                             package="RangesTutorial2017"))
```

Select for deletions:

```
truth <- subset(truth, SVTYPE=="DEL")
calls <- subset(calls, SVTYPE=="DEL")
```

Data cleaning

Make the seqlevels compatible:

```
seqlevelsStyle(calls) <- "NCBI"
truth <- keepStandardChromosomes(truth,
                                 pruning.mode="coarse")
```

Tighten

Move from the constrained VCF representation to a range-oriented model (*VRanges*) with a tighter cognitive link to the problem:

```
calls <- as(calls, "VRanges")
truth <- as(truth, "VRanges")
```

More cleaning

Homogenize the ALT field:

```
ref(truth) <- "."
```

Remove the flagged calls with poor read support:

```
calls <- calls[called(calls)]
```

Comparison

- How to decide whether a call represents a true event?
- Ranges should at least overlap:

```
hits <- findOverlaps(truth, calls)
```

- But more filtering is needed.

Comparing breakpoints

Compute the deviation in the breakpoints:

```
hits <- as(hits, "List")
call_rl <- extractList(ranges(calls), hits)
dev <- abs(start(truth) - start(call_rl)) +
    abs(end(truth) - end(call_rl))
```

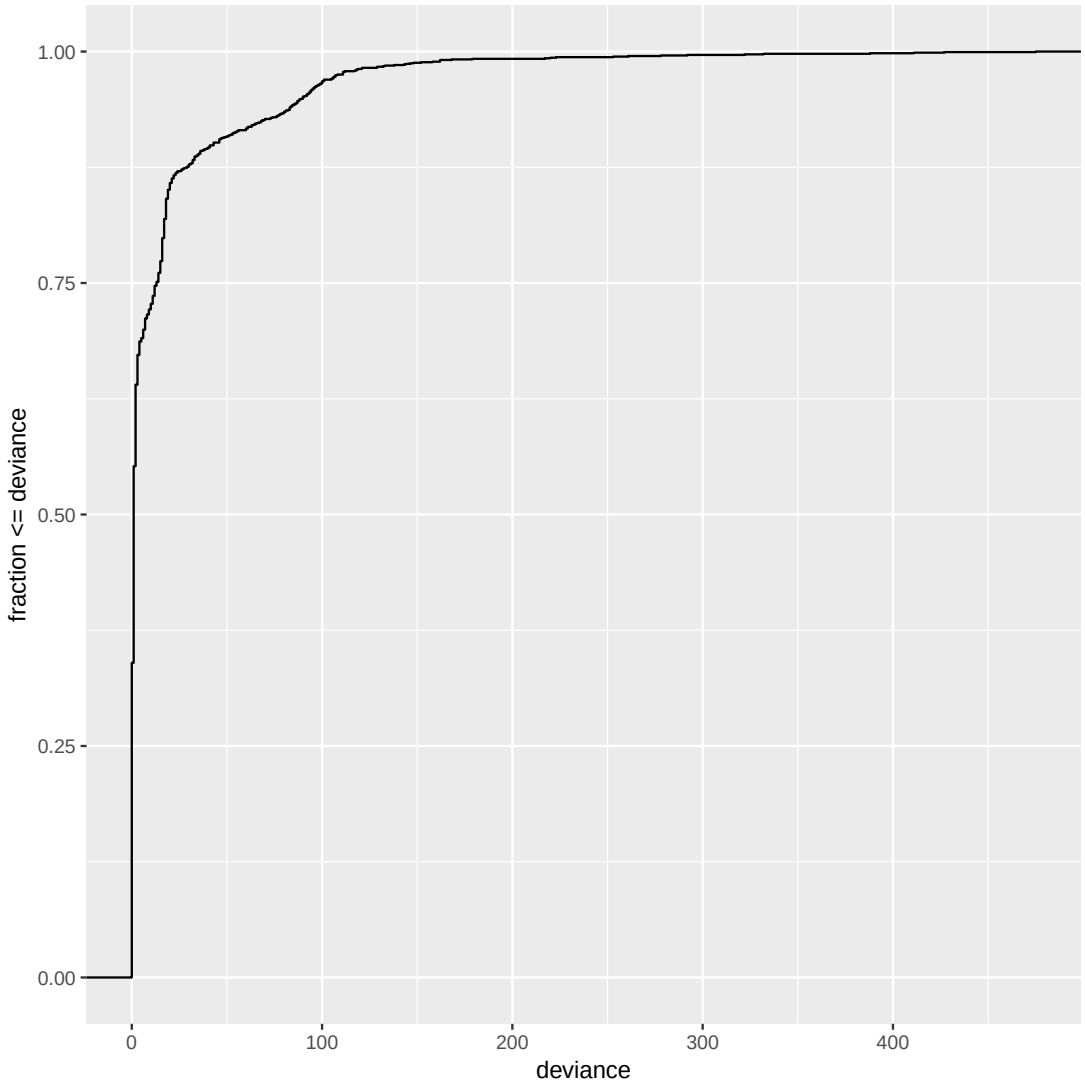
Select and store the call with the least deviance, per true deletion:

```
dev_ord <- order(dev)
keep <- phead(dev_ord, 1L)
truth$deviance <- drop(dev[keep])
truth$call <- drop(hits[keep])
```

Choosing a deviance cutoff

```
library(ggplot2)
rdf <- as.data.frame(truth)
ggplot(aes(x=deviance),
       data=subset(rdf, deviance <= 500)) +
    stat_ecdf() + ylab("fraction <= deviance")
```

Choosing a deviance cutoff



Applying the deviance filter

```
truth$called <- with(truth,
                     !is.na(deviance) & deviance <= 300)
```

Sensitivity

```
mean(truth$called)
```

```
0.82
```

Specificity

Determine which calls were true:

```
calls$fp <- TRUE
calls$fp[subset(truth, called)$call] <- FALSE
```

Compute FDR:

```
mean(calls$fp)
```

```
0.10
```

FDR and variable "alt" regions

- Suspect that calls may be error-prone in regions where the population varies
- Load alt regions from a BED file:

```
bed <-
    system.file("extdata", "altRegions.GRCh38.bed.gz",
                package="RangesTutorial2017")
altRegions <- import(bed)
seqlevelsStyle(altRegions) <- "NCBI"
altRegions <-
    keepStandardChromosomes(altRegions,
                            pruning.mode="coarse")
```

FDR highly associated with "alt" regions

Compute the association between FP status and overlap of an alt region:

```
calls$inAlt <- calls %over% altRegions
xtabs(~ inAlt + fp, calls)
```

inAlt	fp:FALSE	fp:TRUE
FALSE	1402	112
TRUE	58	52