An Introduction to *SNVsOmuC*

Jeremiah Degenhardt

July 10, 2012

Contents

1 Introduction

This vignette outlines the basic usages of the *SNVsOmuC*package and the general workflow for loading data, calling single sample variants and tumor-specific somatic mutations or other sample-specific variant types (eg RNA editing). This package assumes that you have an appropriately mapped BAM file.

2 Calling single-sample vairants

A convenience method is provided to call variants across the entire genome given a BAM file as input. This section assumes that you have installed gmapR and built a genome iit file.

The gmapR package has a function which converts the tally format from gmap (similar to samtools pileup) into a variant GRanges object. This object contains all of the information needed to call variants. The convience function below combines the gmapR tally function with the variant calling functionality here to simultaneously tally the BAM file and call the variants.

```
> library(SNVsOmuC)
> bam_file = file.path(system.file(package = "SNVsOmuC", mustWork=TRUE),
+   "extdata/merged/SRC111111.TESTPREPENDSTR.merged.uniques.bam")
> genome = "hg19_ucsc"
> genome_dir = "/gnet/is2/research/data/bioinfo/gmap/data/genomes"
> variantsGR <- callVariantsP(bam=bam_file, genome=genome,
+ genome_dir=genome_dir,mapq=13, force_qual=TRUE)
> variantsGR$filtered_granges
```

```
GRanges with 2 ranges and 16 elementMetadata cols:
      seqnames             ranges strand |      location         ref
         <Rle>          <IRanges>  <Rle> |   <character> <character>
  [1]    chr12 [6647109, 6647109]      + | chr12:6647109           T
  [2]    chr12 [6647453, 6647453]      + | chr12:6647453           G
             alt   ncycles ncycles.ref     count count.ref count.total
      <character> <integer>   <integer> <integer> <integer>   <integer>
  [1]           C        25          30        30        36          66
  [2]           T         2           0         2         0           2
      high.quality high.quality.ref mean.quality mean.quality.ref
         <integer>        <integer>    <numeric>        <numeric>
  [1]           30               36           71               71
  [2]            2                0           71             <NA>
      count.pos count.pos.ref count.neg count.neg.ref
      <integer>     <integer> <integer>     <integer>
  [1]        17            21        13            15
  [2]         2             0         0             0
  ---
  seqlengths:
                   chr1                 chr10 ...                  chrY
              249250621             135534747 ...              59373566
```

We'll now move on to calling variants from the raw tallied positions in case you have not successfully installed the gmapR package so that I can make some plots.

Below we'll call variants from a single chromosome of a tumor sample and a normal sample so that we can find some somatic mutations.

```
> ##tallies_gr <-get(load(file.path(system.file(package = "SNVsOmuC", mustWork=TRUE),
> ##"extdata/Normal.variant_granges.RData")))
> ##normal <- variantFilter(granges=tallies_gr, useQual=TRUE)
>
> ##tallies_gr <-get(load(file.path(system.file(package = "SNVsOmuC", mustWork=TRUE),
> ##"extdata/Tumor.variant_granges.RData")))
> ##tumor <- variantFilter(granges=tallies_gr, useQual=TRUE)
>   tumor <- list()
> normal <- list()
> tumor$filtered_granges<- variantsGR$filtered_granges
> normal$filtered_granges<- variantsGR$filtered_granges
> normal$raw_granges <-variantsGR$raw_granges
>
```

3 Calling single-sample vairants

next we want to compare the tumor and normal samples to find the tumor-specific mutations...

however, before doing so we need to know that the two samples are actually from the same patient

3.1 Checking that we have data from the same person

```
> concord <- variantConcordance(tumor$filtered_granges, normal$filtered_granges)
> concord
```

```
[1] 1 1
```

now that we know we have tumor and normal data from the same sample we need one more bit of data before we get the tumor-specific variants

```
> normal_cov <- getCov(bam_file)
```

3.2 Actually calling the sample-specific variants

```
> ##tallies_gr <-get(load(file.path(system.file(package = "SNVsOmuC", mustWork=TRUE),
> ##"extdata/Normal_cov.RData")))
> TS<- tumorNormalCompare(tumor_gr=tumor$filtered_granges,
+ normal_gr=normal$filtered_granges, normal_raw=normal$raw_granges,
+ normal_cov=normal_cov)
```

```
Insuficient power in normal 0
```

3.3 Writing a vcf file

Next we want to write out our tumor-specific varints out to a vcf file

```
> cgpGr2vcf(TS, file = "~/temp_ts.vcf", sample_id="Test_sam", project = "SNVsOmuC_Vignette")
```

```
[1] -1
```

4 Making some plots

Finally we want make some plots to check out our data.

The first plot shows us the mutation transition/transversion rate matrix

```
> #plotTitv(TS, main = "tumor specific mutations")
```

and finally we want to plot our variants on the genome

```
> #plotTumor(TS, tumor$filtered_granges)
```