EMBL predoc course 2019 Day1 Delly

Tobias Rausch, modified by Hyobin Jeong

October 21, 2019

Contents

1 Introduction

In this practical we will learn how to identify somatic structural variations using whole genome sequencing of bulk samples (WGS) and Delly (Rausch et al. 2012). We will start from the bam file which includes three example chromosomes (chr10, chr13, chr22) and then make somatic SV calling.

This practical is largely based on the tutorial from Tobias Rausch for SV calling

https://tobiasrausch.com/courses/vc/sv/

2 Basics of linux command

We will firstly start warming up with some basic linux command lines we will frequently use during the practical session.

```
## login to the EMBL server
ssh jeong@seneca

## Go to the specific folder
cd /scratch/

## Make your own folder for the practice session
mkdir name_abcd

## Go to the folder we just made
cd name_abcd

## Check the location where are we now?
pwd

##Here are the list of file paths of the data we want to analyze today
##Let's check how the bam files are look like inside

##1. RPE1 WT
samtools view \
/scratch/jeong/WGS/Practice_RPE/RPEWtp30_chr10_chr13_chr22.bam \
| less -S


##2. RPE1 BM510
samtools view \
/scratch/jeong/WGS/Practice_RPE/BM510_chr10_chr13_chr22.bam \
| less -S

##3. synthetic mixture of WT and BM510 data
samtools view \
/scratch/jeong/WGS/Practice_RPE/BM510_WT_chr10_chr13_chr22_mixture_RG.bam \
| less -S
```

3 Quality Checking of bam file

Paired-end methods can be affected by a skewed insert size distribution, read-depth methods by non-uniform coverage and split-read methods suffer from high sequencing error rates that cause mis-mappings. Prior to any structural variant discovery you should therefore evaluate the quality of the data such as the percentage of mapped reads, singletons, duplicates, properly paired reads and the insert size and coverage distributions. Picard, SAMtools, FastQC and Alfred compute some of these alignment statistics as shown below.

Regarding the QC interpretation, there are some general things to watch out for such as mapping percentages below 70 percentage, larger than 20 percentage duplicates or multiple peaks in the insert size distribution. Be aware that many alignment statistics vary largely by protocol and hence, it's usually best to compare multiple different sequencing runs using the same protocol (DNA-seq, RNA-seq, ChIP-seq, paired-end, single-end or mate-pair) against each other, which then highlights the outliers.

```
alfred qc -r /g/solexa/bin/genomesNew/GRCh38Decoy/GRCh38Decoy.fa \
-o qc_RPEWtp30_chr10_chr13_chr22.tsv.gz -j qc_RPEWtp30_chr10_chr13_chr22.json.gz \
/scratch/jeong/WGS/Practice_RPE/RPEWtp30_chr10_chr13_chr22.bam

zcat qc_RPEWtp30_chr10_chr13_chr22.tsv.gz | grep ^ME \
> qc_RPEWtp30_chr10_chr13_chr22.txt

awk '
{
    for (i=1; i<=NF; i++)  {
        a[NR,i] = $i
    }
}
NF>p { p = NF }
END {
    for(j=1; j<=p; j++) {
        str=a[1,j]
        for(i=2; i<=NR; i++){
            str=str" "a[i,j];
        }
        print str
    }
}' qc_RPEWtp30_chr10_chr13_chr22.txt | less -S
```

Questions for quality checking 1) What is the median coverage of the data set? 2) What is the mapping percentage of the data set? and what will be the cutoff? 3) What is the duplicate rate? and what will be the cutoff?

4 Structural Variant Calling

This is to identify structural variants from normal or tumor samples. This example is to detect duplication (DUP). Let's detect other classes of structural variations also. You can detect several SV types (DEL, DUP, INV, TRA) using the similar command line with -t option.

```
##Example of duplication calling
delly call -n -q 20 -t DUP -g /g/solexa/bin/genomesNew/GRCh38Decoy/GRCh38Decoy.fa \
-o sv_DUP_chr10_chr13_chr22.bcf /scratch/jeong/WGS/Practice_RPE/BM510_chr10_chr13_chr22.bam \
/scratch/jeong/WGS/Practice_RPE/RPEWtp30_chr10_chr13_chr22.bam

##Check the result of duplication calling
bcftools view sv_DUP_chr10_chr13_chr22.bcf | less -S
```

5 Somatic Filtering

This is to identify structural variants from normal or tumor samples. This example is to detect duplication (DUP). Let's detect other classes of structural variations also.

```
##Preparation of sample list
cp /scratch/jeong/WGS/spl.tsv ./


##Example of somatic duplication filtering
delly filter -t DUP -p -f somatic -o somatic_DUP.bcf -a 0.25 \
-s spl.tsv sv_DUP_chr10_chr13_chr22.bcf

##Check the result of somatic duplication filtering
bcftools view somatic_DUP.bcf | less -S
```

Question: What kind of somatic structural variation did you detect in BM510 compared to WT?

6 Visualize SVs in circos plot

In this section, we will visualize detected somatic SV in circos plot using R studio.

```
##Open the R studio
seneca.embl.de

##We need RCircos package for plotting
install.packages('RCircos')

##Initialize parameters
library('RCircos')
data(RCircos.Histogram.Data)
data(RCircos.Heatmap.Data)
data(RCircos.Link.Data)
data("UCSC.HG38.Human.CytoBandIdeogram")

chr.exclude <- "chrY";
cyto.info <- UCSC.HG38.Human.CytoBandIdeogram;
tracks.inside <- 10;
tracks.outside <- 0;
RCircos.Set.Core.Components(cyto.info, chr.exclude, tracks.inside, tracks.outside);
```

```
rcircos.params <- RCircos.Get.Plot.Parameters();
rcircos.cyto <- RCircos.Get.Plot.Ideogram();
rcircos.position <- RCircos.Get.Plot.Positions();
RCircos.List.Plot.Parameters()

rcircos.params <- RCircos.Get.Plot.Parameters();
rcircos.params$heatmap.color <- colorRampPalette(rev(RColorBrewer::brewer.pal(11,"RdBu")))(256);
RCircos.Reset.Plot.Parameters(rcircos.params);
RCircos.List.Plot.Parameters();

##Plot ideogram
out.file <- "RCircosDemoHumanGenome.pdf";
pdf(file=out.file, height=8, width=8, compress=TRUE);
RCircos.Set.Plot.Area();
RCircos.Chromosome.Ideogram.Plot();
dev.off()
```

Now we can add the translocation detected by Delly

```
##Plot translocation using ribbon
data(RCircos.Ribbon.Data)
TL_link <- RCircos.Ribbon.Data[1,]
TL_link$chromA <- c("chr22")
TL_link$chromStartA <- c(37934425)
TL_link$chromEndA <- c(37934425)
TL_link$chromB <- c("chr13")
TL_link$chromStartB <- c(20412250)
TL_link$chromEndB <- c(20412250)

out.file <- "RCircosDemoHumanGenome.pdf";
pdf(file=out.file, height=8, width=8, compress=TRUE);
RCircos.Set.Plot.Area();
RCircos.Chromosome.Ideogram.Plot();

track.num <- 1;
RCircos.Link.Plot(TL_link, track.num, TRUE);
RCircos.Ribbon.Plot(ribbon.data=TL_link, track.num=9, by.chromosome=TRUE, twist=FALSE);
dev.off()
```

7 Detection of subclonal SVs

So far, we have tried to identify clonal SVs in BM510 compared to WT. What about subclonal SVs? To see if we can detect subclonal SVs with low variant allele frequency we have prepared synthetic mixture data which includes 80 percentage of reads from WT and 20 percentage of reads from BM510. If you repeat Delly SV call and somatic filtering on translocation (TRA), can you detect this subclonal SVs which is carried by 20 percentage of the cells?

```
##Call TRA using mixture data
delly call -n -q 20 -t TRA -g /g/solexa/bin/genomesNew/GRCh38Decoy/GRCh38Decoy.fa \
-o sv_TRA_chr10_chr13_chr22_mixture.bcf \
/scratch/jeong/WGS/Practice_RPE/BM510_WT_chr10_chr13_chr22_mixture_RG.bam \
/scratch/jeong/WGS/Practice_RPE/RPEWtp30_chr10_chr13_chr22.bam

##Check the result of TRA calling
bcftools view sv_TRA_chr10_chr13_chr22_mixture.bcf | less -S

##Filtering somatic TRA using mixture data
delly filter -t TRA -p -f somatic -o somatic_TRA_mixture.bcf -a 0.05 \
-s ../spl.tsv sv_TRA_chr10_chr13_chr22_mixture.bcf

##Check the result of somatic TRA calling
bcftools view somatic_TRA_mixture.bcf | less -S
```

8 Session Info

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

- R version 3.5.1 (2018-07-02), `x86_64-apple-darwin15.6.0`
- Locale: `C`
- Running under: `macOS High Sierra 10.13.3`
- Matrix products: default
- BLAS: `/System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/libBL`
- LAPACK: `/Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.26.0, GenomeInfoDb 1.16.0, GenomicRanges 1.32.7, IRanges 2.14.12, RCircos 1.2.1, S4Vectors 0.18.3, breakpointR 1.0.0, breakpointRdata 1.0.0, cowplot 0.9.4, ggplot2 3.1.1, knitr 1.20
- Loaded via a namespace (and not attached): Biobase 2.40.0, BiocParallel 1.14.2, BiocStyle 2.8.2, Biostrings 2.50.2, DT 0.4, DelayedArray 0.6.6, DelayedMatrixStats 1.2.0, FNN 1.1.2.1, GenomeInfoDbData 1.1.0, GenomicAlignments 1.16.0, MOFAtools 0.99.0, Matrix 1.2-14, MultiAssayExperiment 1.6.0, R6 2.4.0, RColorBrewer 1.1-2, RCurl 1.95-4.12, Rcpp 1.0.1, Rhdf5lib 1.2.1, Rsamtools 1.34.0, SingleCellExperiment 1.2.0, SummarizedExperiment 1.10.1, XVector 0.20.0, assertthat 0.2.1, backports 1.1.4, beeswarm 0.2.3, bindr 0.1.1, bindrcpp 0.2.2, bitops 1.0-6, codetools 0.2-15, colorspace 1.4-1, compiler 3.5.1, corrplot 0.84, crayon 1.3.4, data.table 1.12.2, digest 0.6.19, doParallel 1.0.14, dplyr 0.7.6, dynamicTreeCut 1.63-1, edgeR 3.22.3, evaluate 0.11, foreach 1.4.4, ggbeeswarm 0.6.0, ggrepel 0.8.0, glue 1.3.1, grid 3.5.1, gridExtra 2.3, gtable 0.3.0, gtools 3.8.1, highr 0.7, htmltools 0.3.6, htmlwidgets 1.2, httpuv 1.5.1, igraph 1.2.2, iterators 1.0.10, jsonlite 1.6, later 0.8.0, lattice 0.20-35, lazyeval 0.2.2, limma 3.36.3, locfit 1.5-9.1, magrittr 1.5, matrixStats 0.54.0, mime 0.6, munsell 0.5.0, pheatmap 1.0.10, pillar 1.4.0, pkgconfig 2.0.2, plyr 1.8.4, promises 1.0.1, purrr 0.2.5, reshape2 1.4.3, reticulate 1.10, rhdf5 2.24.0, rjson 0.2.20, rlang 0.3.4, rmarkdown 1.10, rprojroot 1.3-2, rstudioapi 0.7, scales 1.0.0, scater 1.8.4, scran 1.8.4, shiny 1.3.2, shinydashboard 0.7.0, statmod 1.4.30, stringi 1.4.3, stringr 1.4.0, tibble 2.1.1, tidyr 0.8.1, tidyselect 0.2.4, tinytex 0.7, tools 3.5.1, tximport 1.8.0, vipor 0.4.5, viridis 0.5.1, viridisLite 0.3.0, withr 2.1.2, xfun 0.3, xtable 1.8-4, yaml 2.2.0, zlibbioc 1.26.0