Bioinformatics Practical Part 3

Discovering structural variants from mapped Oxford Nanopore reads

MP235

SS 2022

Introductory information.

Up till now we learned about read mapping and worked with SAM files. Reads mapped to the genome can be used for structural variant discovery. Structural variants are variants over >50 bp in length. The most commonly found SVs are insertions and deletions. When we talk about genome re-sequencing for SV discovery one often hears the term genome coverage (for example 5x, 20x, 20x).

Question 1

What is genome coverage? How does genome coverage affect SV discovery? (Refer to this manuscript: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04422-y)

A file that will be used for SV calling is found in the /mnt/practical2022/part3/files folder. Move it to your work directory:

```
$ mkdir /mnt/practical2022/part3/analysis
$ mv /mnt/practical2022/part3/files/SRR15731030.minimap2.20x.bam* \
> /mnt/practical2022/part3/analysis
$ mv /mnt/practical2022/part2/analysis/Express617_v1.fa \
> /mnt/practical2022/part3/analysis
$ cd /mnt/practical2022/part3/analysis
```

Question 2

Why do we use a star in the mv command?

BAM file QC

Have a look at the header.

Question 3

Is the file coordinate sorted?

Now let's check the mapping statistics

```
$ samtools flagstat -@ 8 SRR15731030.minimap2.20x.bam
```

Question 4

What is `-@ 8` in the command?

Question 5

How many reads are mapped?

Question 6

What are secondary and supplementary alignments? Check in the samtools specifications: https://samtools.github.io/hts-specs/SAMv1.pdf

Let's check depth for a small region

```
$ samtools depth -r chrA01:1-1000000 SRR15731030.minimap2.20x.bam > \
> chrA01:1-1000000.depth
```

Question 7

What is the approximate depth over that region?

Structural Variant calling

There are several tools, which can be used to call structural variants from a bam file.

Question 8

List at least three tools for SV discovery from mapped Oxford Nanopore reads.

We will use cuteSV. Have a look at the github page (https://github.com/tjiangHIT/cuteSV) and at the associated manuscript:
https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-02107-y

Type `cuteSV` into your terminal.

Question 9

What do you see?

Now type `cuteSV -h` into your terminal.

Question 10

What do you see? Sometimes you need to type `-h`, `-help` or `-help` to get the full help information.

Now try to construct a cuteSV command to call variants. Remember you are using Oxford Nanopore reads, so adjust parameters accordingly. Use the help displayed on the terminal or the github page to help you.

```
$ cuteSV -t 8 -s 8 --genotype --max_cluster_bias_INS 100 \
> --diff_ratio_merging_INS 0.3 --max_cluster_bias_DEL 100 \
> --diff_ratio_merging_DEL 0.3 SRR15731030.minimap2.20x.bam \
```

```
> Express617_v1.fa cutesv.vcf cutesv.out
```

Question 11

What is the meaning of the different parameters? Why are we using these parameters? Consult the cuteSV github page.

Question 12

What format are the variants stored in?

Let's have a look at the VCF files specification:
https://samtools.github.io/hts-specs/VCFv4.2.pdf

Question 13

How many columns are there in a VCF file, what do they mean?

The cuteSV command will take a long time (several hours) to run. You can stop it using ctrl+c (strg+c). There is a ready output file (prapared with the same command) in the /mnt/practical2022/part3/svs directory.

```
$ cd /mnt/practical2022/part3/svs
$ ll cutesv.vcf
```

Let's get a summary for the structural variants. We will use a custom python script to do that (feel free to have a look inside.)

```
$ python /mnt/practical2022/part3/scripts/get_summary.py cutesv.vcf
```

Question 14

How many variants are there? What types? How many insertions? How many deletions?

Question 15

What is the meaning of homozygous/heterozygous?

In the directory there is a second file sniffles.vcf which was produced by a different SV calling software (Sniffles2) using a command below **(no need to run it!)**

```
$ sniffles --minsupport 8 --threads 8 --input SRR15731030.minimap2.20x.bam \
> --vcf sniffles.vcf
```

Let's get a summary for the structural variants using the custom script again.

```
$ python /mnt/practical2022/part3/scripts/get_summary.py sniffles.vcf
```

Question 16

How many variants are there? What types? How many insertions? How many deletions?

Now let's compare variants obtained from cuteSV and Sniffles2. For the downstream comparisons to work we need to slightly re-format the VCF files (output files from different SV discovery software all conform to VCF format, but have some differences). Besides re-formatting, the script does filtering, keeping only variants fulfilling the following criteria:

- Minimum SV length: 50 bp
- SV type: insertion or deletion
- SV quality: SVs flagged as "PASS"
- genotype: homozygous genotype for alternative allele ('1/1')

```
$ python /mnt/practical2022/part3/scripts/bare_bones.py \
> bare.bones.header.bn cutesv.vcf > cutesv.bb.vcf
$ python /mnt/practical2022/part3/scripts/bare_bones.py \
> bare.bones.header.bn sniffles.vcf > sniffles.bb.vcf
```

Lets get variant summaries again.

```
$ python /mnt/practical2022/part3/scripts/get_summary.py cutesv.bb.vcf
$ python /mnt/practical2022/part3/scripts/get_summary.py sniffles.bb.vcf
```

Question 17

Are there any differences?

Let's compare variant calls. It can be done using `surpyvor`. Type `surpyvor` into your terminal.

Question 18

What sub-commands are there?

Let's get intersection of SVs

Question 19

What options does the command `surpyvor highconf` have?

Question 20

Why is the command which gives intersection called `highconf`?

```
$ surpyvor highconf -o intersection.vcf --variants cutesv.bb.vcf sniffles.bb.vcf
```

Let's get stats for our high confidence set.

```
$ python /mnt/practical2022/part3/scripts/get_summary.py intersection.vcf
```

Question 21

How do they compare with individual files?

Let's overlap variants with genes. We can achieve that using `bedtools`. GFF files normally will have entries for genes, transcripts (mRNAs) and exons (possibly also some other features). Features are listed in column 3.

You can find out which features are present with this command.

```
$ grep -v "^#" Express617_v1_gene.gff3 | cut -f 3 | sort | uniq
```

Question 22

Which features do we have in this gff3 file?

Let's keep only the mRNA features

```
$ grep -P "\tmRNA\t" Express617_v1_gene.gff3 > Express617_v1_gene.mRNA.gff3
```

Type `bedtools` into your terminal.

We are interested in the `bedtools intersect` function.

```
$ bedtools intersect -nonamecheck -wa -a intersection.vcf \
> -b Express617_v1_gene.mRNA.gff3 > SV.mRNA.overlap.tsv
```

What other arguments are there. Try `bedtools intersect` with other sets of arguments.

Dataset for the final report

Choose one variant overlapping a gene from SV.mRNA.overlap.tsv. Note the variant ID. Use `samtools` to extract a 1 Mb region around the variant. Refer to the final report instructions for further information.