WTAC NGS Bioinformatics Course

Module 5: Structural variant calling

Open a terminal and go to the Module 5 directory.

Exercise 1: VCF for SVs

On the terminal, cd to the 'exercise1' directory.

There is a VCF file called ERR1015121.vcf that was produced using the Lumpy SV calling software. You can read the VCF file by using the less command:

```
less ERR1015121.vcf
```

- What does the CIPOS format tag indicate?
- What does the PE tag indicate?
- What tag is used to describe an inversion event?
- What tag is used to describe a duplication event?
- How many deletions were called in total? (Hint: DEL is the info field for a deletion. The -c option of the grep command can be used to return a count of matches.)
- What type of event is predicted at IV:437148? What is the length of the SV? How many paired-end reads and split-reads support this SV variant call?
- What is the total number of SV calls predicted on the IV chromosome?

Exercise 2: Breakdancer

On the terminal, cd to the 'exercise2' directory.

In this exercise, we will use the Breakdancer software package to call structural variants on a yeast sample that was paired-end sequenced on the illumina HiSeq 2000.

2.1 Breakdancer first needs to examine the BAM file to get information on the fragment size distribution for each sequencing library contained in the BAM file.

The `breakdancer.config` file has information about the sequencing library fragment size distribution. Use the 'cat' command to print the contents of the '`breakdancer.config`' file.

- What is the mean and standard deviation of the fragment size?

2.2 Now we will run the breakdancer SV caller. Run the command:

```
breakdancer-max breakdancer.config > ERR1015121.breakdancer.out
```

The output from Breakdancer is a simple text format with one line per SV event (**NOT** VCF format). Breakdancer calls four different types of structural variants: deletions (DEL), insertions (INS), inversions (INV), intra chromosomal translocations (ITX), and inter chromosomal translocations (CTX).

- What type of SV event is predicted at position III:83065? What is the size of this SV? What is the score of this SV?
- What type of SV event is predicted at position II:258766?

2.3 Next you will convert the output of breakdancer into a standard format call BED, that is accepted by many other tools and genome browsers. The BED format is explained here: https://genome.ucsc.edu/FAQ/FAQformat.html#format1

Can you create a BED file for the **deletions** that were predicted by breakdancer in 2.2 using standard unix commands. Here are some hints:

1. Extract all the deletions from the breakdancer.out file (Hint: use grep) 2. We want to print columns: 1, 2, 5, 7, and 9 to correspond to create a BED file with columns: chromosome, start, end, name, and score. (Hint: use awk to do this, e.g. awk '{print $1"\t"$2}')
3. Print the resulting bed output into a file called: breakdancer.dels.bed

2.4 Can you open the IGV genome browser and inspect the SV event at II:258766. To do this, type:

```
igv.sh &
```

First, you need to open the genome (click on the leftmost dropdown box at the top, and find 'S. cerevisiae EF3 r62' and select it). Then use the File menu to open your BAM file (File - Load from file, and select ERR1015121.bam). Next load the BED file for the deletion calls that you created in 2.3 (File - Load from file, and select 'breakdancer.dels.bed').

In the location text box, type: II:257766-259766:

- Can you see the structural variant? (Hint: you may need to zoom out a little to see the full structural variant).
- Can you see any evidence to support this SV call?
- Can you estimate the size of the SV?

The VCF in the exercise 1 directory was produced by another structural variant caller on the same sample as this exercise. Can you load this VCF into IGV also (File - Load from file, and select ERR1015121.vcf in the exercise 1 directory) and answer the following questions:

- Was the deletion at II:258766 also called by the other structural variant software?
- Can you navigate to II:508,064-511,840? Is there a SV deletion called in this region by either SV caller? Is there any read support for a SV deletion in this region? If so, how many read pairs could support the deletion call (Hint: change the IGV view to 'squished' and 'View as pairs' to see any inconsistently aligned read pairs).

Exercise 3: Lumpy structural variant caller

In this exercise, we will use the Lumpy software package to call structural variants on a yeast sample that was paired-end sequenced on the Illumina Hiseq 2000.

Lumpy is designed to take BAM files that have been aligned with BWA-mem.

3.1 On the terminal, cd to the 'exercise3' folder and check that there is a BAM file called ERR1015069.bam and index file in the directory (hint: ls -l).

3.2 The first step for running Lumpy is to extract the read pairs that are discordantly mapped (i.e. pairs that are not mapped within the expected fragment size distribution). We will use Samtools to extract these reads:

```
samtools view -bh -F 1294 ERR1015069.bam | samtools sort -O bam -T
ERR1015069.temp -o ERR1015069.discordants.bam
```

- Can you index the bam file? (Hint: use samtools index)
- What does the -F option in 'samtools view' do?
- Which BAM flags does 1294 indicate? (Hint: in your web browser, visit https://broadinstitute.github.io/picard/explain-flags.html and enter 1294 to find out)

3.3 Next we will use Lumpy to extract the reads that are only split mapped (i.e. split read alignments). This is all one single command:

```
samtools view -h ERR1015069.bam | extractSplitReads_BwaMem -i stdin | samtools
view -b - | samtools sort -O bam -T ERR1015069.temp -o ERR1015069.splitters.bam
```

- Can you index the bam file? (Hint: use samtools index)

3.4 We will now do the structural variant calling with lumpy, providing it with the original BAM file and the two BAM files we prepared in 3.2 and 3.3.

```
lumpyexpress -B ERR1015069.bam -S ERR1015069.splitters.bam -D
ERR1015069.discordants.bam -o ERR1015069.vcf
```

- What type of SV event occurs at position IV:383993? What is the length of the SV event?
- What type of SV event occurs at position XV:43018? What is the length of the SV event?

Exercise 4: Sniffles - long read SV caller

Sniffles is a SV caller that is designed for long reads (Pacbio or Oxford Nanopore). It is very important that the reads are first aligned with an aligner suitable for long reads. NGMLR is a long-read mapper designed to align PacBio or Oxford Nanopore (standard and ultra-long) to a reference genome with a focus on reads that span structural variations.

We will use data from a *Saccharomyces cerevisiae* strain (YPS128) that was sequenced at the Wellcome Trust Sanger Institute and deposited in the ENA (Project: PRJEB7245, sample: SAMEA2757770, analysis: ERZ448241).

4.1 On the terminal, cd to the 'exercise4' directory.

The sequencing reads are contained in a fastq file:
`YPS128.filtered_subreads.10x.fastq.gz`

The reference genome is in the ref directory in a fasta file:
`Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa`

- Can you align the reads with NGMLR and send the output to a SAM file called `YPS128.10x.filtered_subreads.sam`? You can find the usage of ngmlr by typing: `ngmlr`

Note: the -t parameter to use multiple threads in parallel (this will increase the speed of the alignment by using more than one CPU core - for these machines, I suggest using 6).

- Can you convert the output to BAM format (`samtools view -b`)
- Sort the BAM file (`samtools sort`) and produce a sorted BAM file called: `YPS128.10x.filtered_subreads.sorted.bam`
- Finally, use samtools to index the sorted BAM file (`samtools index`).

4.2 We will now use the BAM file to do structural variant calling with Sniffles.

Sniffles takes the BAM file as input and outputs VCF. Using the default parameters, can you call SVs with Sniffles and output the results into a VCF file called `YPS128.10x.vcf`. To find the usage for Sniffles, type:

```
sniffles
```

You don't need to change any of the default parameters, but you will need to work out how to provide the input BAM file and specify the output VCF file. The documentation on sniffles is here: https://github.com/fritzsedlazeck/Sniffles/wiki/Parameter

4.3 IGV inspection

- What sort of SV was called at on chromosome 'Mito' at position 29295?
- What is the length of the SV?
- How many reads are supporting the SV?
- From a visual inspection of the SV in IGV, can you determine how accurate is the breakpoint of the called SV compared to what you see in IGV?

Optional exercise 5: Bedtools

On the terminal, cd to the 'exercise5' directory.

Bedtools is an extremely useful tool for doing regional comparisons over genomic co-ordinates. It has many commands for doing region based comparisons with BAM, VCF, GFF, BED file formats. To see the list of commands available, on the command line type:

```
bedtools
```

In this directory, there are two VCF files and the yeast genome annotation in GFF3 format (Saccharomyces_cerevisiae.R64-1-1.82.genes.gff3).

5.1 For instance, we can quickly find out how many of the SVs intersect with annotated regions of the genome by using the `'bedtools intersect'` command. For the intersect command, the -a and -b parameters are used to denote the input files.

- Using the `'bedtools intersect'`, can you determine how many SVs in ERR1015069.dels.vcf overlap with an annotated region of the yeast genome? (hint: use the unix command wc to count the number of lines in the output, and note the -u parameter in bedtools intersect).
- How many SVs do not overlap with a gene? (hint: note the -v parameter to bedtools intersect)

5.2 The default is to report overlaps between features in A and B so long as at least one base pair of overlap exists. However, the `-f` option allows you to specify what fraction of each feature in A should be overlapped by a feature in B before it is reported.

- Can you be more strict and require 50% of overlap between the SV and the genes? ● How many SVs overlap with this more strict definition?

5.3 You can use bedtools to find the closest feature to a SV using the 'bedtools closest' command.

- What is the closest gene to the structural variant at IV:384220?

5.4 We can use the `'bedtools intersect'` command to determine how many SVs overlap between two VCF files.

- Can you determine how many SVs have a 50% reciprocal overlap between the two files: `ERR1015069.dels.vcf ERR1015121.dels.vcf` (Hint: first find the option for reciprocal overlap by typing: bedtools intersect -h)

References:

LUMPY: a probabilistic framework for structural variant discovery
http://www.genomebiology.com/2014/15/6/R84

BreakDancer: an algorithm for high-resolution mapping of genomic structural variation
http://www.nature.com/nmeth/journal/v6/n9/abs/nmeth.1363.html

NGMLR: https://github.com/philres/ngmlr
Sniffles: https://github.com/fritzsedlazeck/Sniffles/wiki
More information can be found in this paper: Accurate detection of complex structural variations using single molecule sequencing
https://www.biorxiv.org/content/early/2017/07/28/169557

BEDTools: a flexible suite of utilities for comparing genomic features
http://bioinformatics.oxfordjournals.org/content/26/6/841.long

URLs

Lumpy-SV: https://github.com/arq5x/lumpy-sv
Breakdancer: http://gmt.genome.wustl.edu/packages/breakdancer/
Bedtools: https://github.com/arq5x/bedtools

Documentation

Calling SVs with Lumpy: https://github.com/arq5x/lumpy-sv#lumpy-traditional-usage