GULBENKIAN SCIENCE

PO: Precision Oncology Course

Variant Detection: OVCA case

BU CNIO BIOINFORMATICS UNIT

cnio stop cancer

Objectives

- Understand how to configure varca
- Run the varca pipeline
- Check out the variant calling results
  - SNVs and indels
  - Germline variants
  - Somatic variants
- Visualize the variants using IGV

The OVCA case

- Patient with ovarian cancer
- Whole exome sequencing from two samples from the patient:
  - Tumor sample
  - Matched normal sample (healthy tissue) from epithelium
- Library: Agilent SureSelect V5 Human All Exons
- Sequencing platform: HiSeq 2000 (Illumina)
- Paired-end sequencing

NOTE: This data was simulated and reduced (only chromosome 17) in order to perform the computational analysis in class time.

The data

Link for download: OVCAcase.zip

Raw_data
WEx_Normal_R1.fastq
WEx_Normal_R2.fastq
WEx_Tumour_R1.fastq
WEx_Tumour_R2.fastq

Raw exome sequencing data
Patient's sample data, generated by the sequencing machine
Source: Collaborator/Sequencing provider

Reference
hg19_chr17.fa

Human reference genome
Consensus genome reference
Source: sequence databases (e.g. UCSC)

Index

Index of reference genome
Directory for the files containing the index for BWA-MEM2 aligner
Source: Will be created after BWA-MEM2 index execution

Annotations
dbsnp_138.hg19_chr17.vcf.gz

dbSNP annotation file
Reported small variants
Source: dbSNP

Intervals
SureSelect_V5_human_all_Exons_chr17.bed

WEx Library design
Predefined genomic regions of interest
Source: manufacturer

Setting up the analysis

1. Make a local copy of the data and check its structure

```
$ cd /home/user/OVCA_case
$ tree
.
├── Annotations
│   └── dbsnp_138.hg19_chr17.vcf.gz
├── Index
├── Intervals
│   └── SureSelect_V5_human_all_Exons_chr17.bed
├── REFERENCE
│   └── hg19_chr17.fa
└── Raw_data
    ├── WEx_Normal_R1.fastq
    ├── WEx_Normal_R2.fastq
    ├── WEx_Tumour_R1.fastq
    └── WEx_Tumour_R2.fastq

5 directories, 7 files
```

Setting up the analysis

2. Configure samples.tsv file

- Use samples-example.tsv inside varca as a template
- Open the file in a text editor
- Modify the template according to the requirements of the analysis:
  - Identification of both somatic and germline alterations -> Execution of MuTect2
  - We have tumor and normal samples -> Execution of MuTec2 in tumor-normal mode
    - MuTect2 execution in tumor-normal mode:

group	sample	control
1	A	B
1	B	-

```
group   sample  control
1       Tumor   Normal
1       Normal  -
~
~
~
```

- Save the changes
- Rename the file from samples-example.tsv to samples.tsv

Setting up the analysis

3. **Configure units.tsv file**
   - Use units-example.tsv inside varca as a template
   - Open the file in a text editor
   - Modify the template according to the requirements of the analysis:
     - We have to keep <u>same nomenclature for the samples</u> in samples.tsv and units.tsv
     - Both samples were sequenced once -> <u>unit is 1 for each of them</u>
     - Sequencing platform was <u>ILLUMINA</u>
     - Sequencing was paired-end -> <u>fq1 and fq2 must be informed</u>

```
sample  unit    platform        fq1     fq2
Tumor   1       ILLUMINA        /home/user/OVCAcase/Raw_data/WEx_Tumour_R1.fastq        /home/user/OVCAcase/Raw_data/WEx_Tumour_R2.fastq
Normal  1       ILLUMINA        /home/user/OVCAcase/Raw_data/WEx_Normal_R1.fastq        /home/user/OVCAcase/Raw_data/WEx_Normal_R2.fastq
~
~
~
```

   - Save the changes
   - Rename the file from units-example.tsv to units.tsv

Setting up the analysis

4. Configure contigs.tsv file

- Use contigs-example.tsv inside varca as a template
- Open the file in a text editor
- Modify the template according to the requirements of the analysis:
  - As we are only analyzing chromosome 17 erase all the chromosomes except chr17
- Save the changes
- Rename the file from contigs-example.tsv to contigs.tsv

Setting up the analysis

5. **Configure config.yaml file**
   - Use config-example.yaml inside varca as a template
   - Open the file in a text editor
   - Modify the template according to the requirements of the analysis:

     (see next slide)
   - Save the changes
   - Rename the file from config-example.yaml to config.yaml

Setting up the analysis

5. **Configure config.yaml file**
    - ref:
        - name: GRCh37.75
        - genome: /home/user/OVCAcase/REFERENCE/hg19_chr17.fa
        - genome_idx: /home/user/OVCAcase/Index/
        - known-variants: /home/user/OVCAcase/Annotations/dbsnp_138.hg19_chr17.vcf.gz
    - processing:
        - restrict-regions: /home/user/OVCAcase/Intervals/SureSelect_V5_human_all_Exons_chr17.bed
    - annotation:
        - vep:
            - cache: true
            - cache_version: 105
            - cache_directory: /home/user/OVCAcase/
            - assembly: GRCh37
            - annotations: "--sift b --polyphen b --ccds --uniprot --hgvs --symbol --numbers --domains --regulatory --canonical --protein --biotype --uniprot --tsl --af --variant_class --xref_refseq --af_1kg --af_esp --af_gnomad --appris --fasta /home/user/OVCAcase/homo_sapiens/105_GRCh37/Homo_sapiens.GRCh37.75.dna.primary_assembly.fa.gz”

Running varca: download vep-cache

- Locate the vep environment

```
(snakemake) $ find .snakemake/conda -name vep
.snakemake/conda/1bad1ac03e4ab74dbd5a7022b3c37b5f/bin/vep
.snakemake/conda/1bad1ac03e4ab74dbd5a7022b3c37b5f/share/ensembl-vep-105.0-1/vep
```

- Activate vep environment

```
(snakemake) $ conda activate .snakemake/conda/1bad1ac03e4ab74dbd5a7022b3c37b5f
```

- Install vep cache files

```
(/storage/scratch01/users/epineiro/varca/.snakemake/conda/
1bad1ac03e4ab74dbd5a7022b3c37b5f) $ vep_install -a cf -c /home/user/OVCAcase/ -y GRCh37
-s homo_sapiens
```

- Deactivate vep environment

```
(/storage/scratch01/users/epineiro/varca/.snakemake/conda/
1bad1ac03e4ab74dbd5a7022b3c37b5f) $ conda deactivate
```

Running varca: partial execution

- First run varca until FastQC step. This will only execute raw data quality control.

```
(snakemake) $ snakemake --use-conda --cores 2 —until fastqc
```

- To execute until the creation of the VCF files (no VEP annotation)

```
(snakemake) $ snakemake --use-conda --cores 2 —until merge_calls
```

Execution of HaplotypeCaller

```
(snakemake) $ snakemake --use-conda --cores 2 —until filter_mutect_2
```

Execution of MuTect2

Running varca: full execution

- To run the full pipeline

```
(snakemake) $ snakemake --use-conda --cores 2
```

- Once finished, generate report (this step only works if all steps have been previously run)

```
(snakemake) $ snakemake --report report.html
```

Where to find the results

```
out
├── annotated
│   ├── 1.snpeff.vcf.gz
│   ├── 1.vep.vcf.gz
│   ├── 1.vep.vcf.gz_summary.html
│   ├── Tumor_mutect.vep.vcf.gz
│   └── Tumor_mutect.vep.vcf.gz_summary.html
├── filtered
│   ├── 1.indels.hardfiltered.vcf.gz.tbi
│   ├── 1.indels.vcf.gz.tbi
│   ├── 1.snvs.hardfiltered.vcf.gz.tbi
│   ├── 1.snvs.vcf.gz.tbi
│   ├── 1.vcf.gz
│   └── 1.vcf.gz.tbi
├── mutect_filter
│   ├── Tumor_passlable.vcf.gz
│   ├── Tumor_passlable.vcf.gz.filteringStats.tsv
│   ├── Tumor_passlable.vcf.gz.tbi
│   ├── Tumor_passlable_filtered.vcf.gz
│   └── Tumor_passlable_filtered.vcf.gz.tbi
├── qc
│   ├── fastqc
│   │   ├── Normal-1-r1_fastqc.html
│   │   ├── Normal-1-r1_fastqc.zip
│   │   ├── Normal-1-r2_fastqc.html
│   │   ├── Normal-1-r2_fastqc.zip
│   │   ├── Tumor-1-r1_fastqc.html
│   │   ├── Tumor-1-r1_fastqc.zip
│   │   ├── Tumor-1-r2_fastqc.html
│   │   └── Tumor-1-r2_fastqc.zip
│
├── recal
│   ├── Normal-1.bai
│   ├── Normal-1.bam
│   ├── Normal-1.grp
│   ├── Tumor-1.bai
│   ├── Tumor-1.bam
│   └── Tumor-1.grp
```

annotated VCF files

VCF file with all HC variants and PASS label

VCF file with all MuTect2 variants and PASS label

FastQC output files

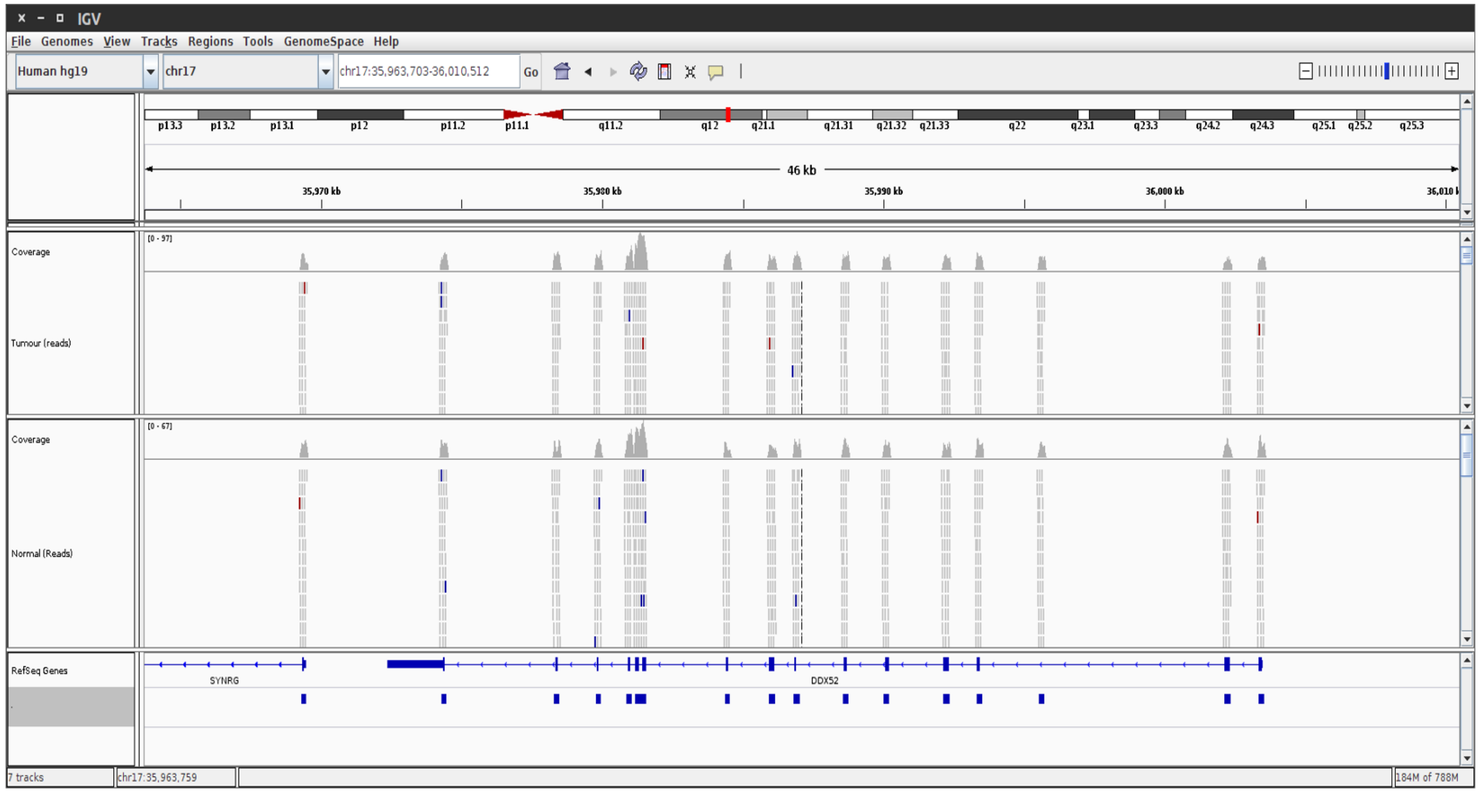
recalibrated alignment bam files

Extract with main results

Alignment Visualization

1. Open a terminal, and execute the command

```
(snakemake) $ conda install -c bioconda igv
(snakemake) $ igv
```



2. Open the BAM files for each sample

tumor-1.bam

normal-1.bam

3. Open the BED file (intervals files)

SureSelect_V5_human_all_Exons_chr17.bed

Manual: https://software.broadinstitute.org/software/igv/UserGuide

Questions

Germline variants:

There were detected ___ germline variants in total:

- ___ Single Nucleotide Variants
- ___ Indels

Somatic variants:

Genes affected and type of mutations (see alignment using IGV on chr17):

- _____
- _____

Consider only those variants with PASS label

Extra: Data formats cheat sheet

Format	Uses	Example	File type	Software Management	File Extension
Fasta	**Human genome** Define biological sequences (DNA, RNA, cDNA, proteins).	human_genome.fa	Plain text	samtools, picard-tools	.fa; .fasta
FastQ	**Raw sequencing data** Single-end sequencing → 1 file Paired-end sequencing → 2 files (R1 and R2 for each end, respectively)	DNAseq_raw_data.fastq (DNAseq_R1.fastq and DNAseq_R2.fastq)	Plain text	samtools, picard-tools **Aligners**	.fq; .fastq
SAM	Define read alignments. Store alignment meta-info (reference, methods, one- or multi-sample).	mapped_reads.sam	Plain text	samtools, picard-tools	.sam
BAM	**VISUALIZE ALIGNMENTS (IGV)** The same as SAM, but compressed and indexed. Also to store UNMAPPED reads (compressed).	mapped_reads.bam unmapped_reads.bam	Binary	samtools, bcftools, picard-tools, **IGV** (Integrative Genome Viewer)	.bam
VCF	**SNV & Indels calls** Indicates genomic variations. Store Variant calling meta-info (reference, methods,one- or multi-sample).	point_variants.vcf	Plain text	bcftools, Unix	.vcf
BED	**Intervals** Delimit genomic regions (i.e. intervals) w or w/o annotations.	targeted_regions.bed intervals.bed	Plain text	bedtools, Unix **GATK, picard-tools**	.bed
TSV or CSV	Create data matrix (rows X Columns)	annotated_variants.tsv	Plain text	Unix, Microsoft Excel, OpenOffice	.tsv; .csv; .txt