GULBENKIAN SCIENCE

PO21: Precision Oncology Course

Panel Analysis - GastroIntestinal Cancer Patient

BU CNIO BIOINFORMATICS UNIT

cnio stop cancer

Study Case - Gastrointestinal Cancer Patient

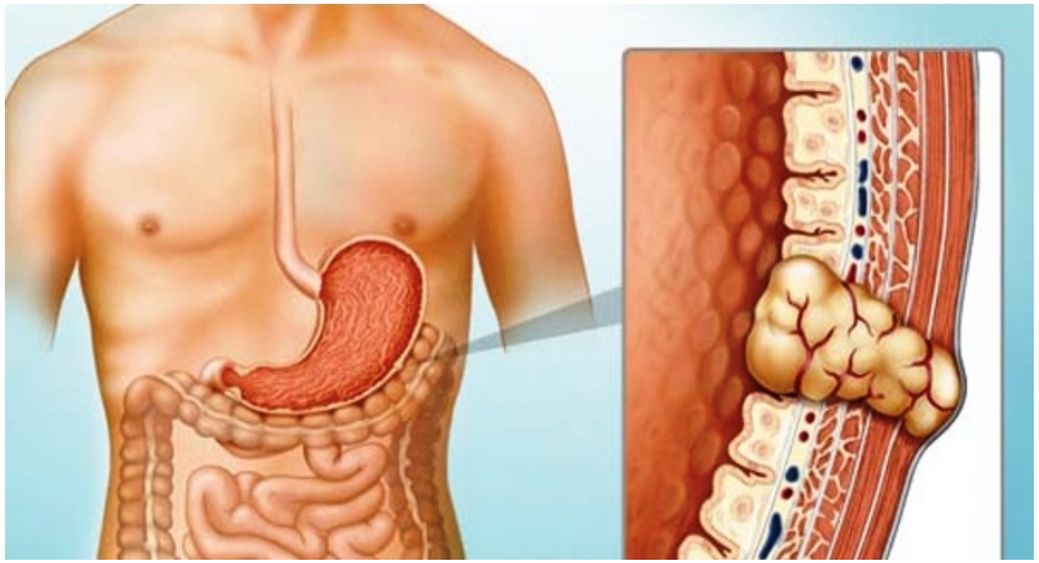
Tumor type: Patient with lung metastasis (gastrointestinal stromal tumor)

Sequencing: Illumina HiSeq2500

Type of data: Sequencing panel (paired). Ion Ampliseq Comprehensive Cancer Panel (409 genes)

Reference Genome: hg19

Samples: One tumor sample without normal tissue



Gastrointestinal Stromal Tumors (GIST), from Mangalore Institute of Oncology.

Study Case - Panel 2

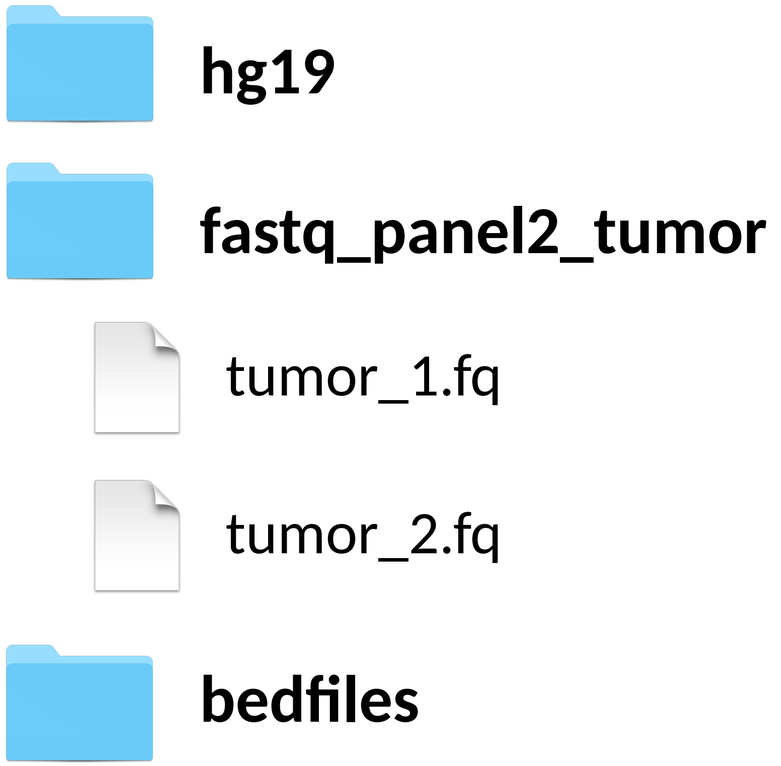
Tumor type: Patient with lung metastasis (gastrointestinal stromal tumor)

Sequencing: Illumina HiSeq2500

Type of data: Sequencing panel (paired). Ion Ampliseq Comprehensive Cancer Panel (409 genes)

Reference Genome: hg19

Samples: One tumor sample without normal tissue



Move to working directory

1. Create a new folder in de Desktop and move to it

```
$ mkdir Desktop/GIST
$ cd Desktop/GIST
```

2. Clone varca

```
$ git clone https://gitlab.com/bu_cnio/varca
```

3. Move inside varca folder

```
$ cd varca
```

Edit configuration files

contigs.tsv

We are leaving this file empty so the analysis is performed using all contigs of the fasta index.

```
$ touch contigs.tsv
```

Edit configuration files

samples.tsv

As we only have tumor sample, MuTect2 will be executed in **tumor-only mode**. In order to do so, the **sample name** must be **the same in the sample and control columns**.

sample	control
tumor	tumor

Rename samples file: `$ mv samples-example.tsv samples.tsv`

Edit configuration files

units.tsv

We only have **one tumor sample** with **two FASTQ files**: read 1 and read2.

sample	unit	platform	fq1	fq2
tumor	1	ILLUMINA	/Users/participant/Desktop/panel2/tumor_1.fq	/Users/participant/Desktop/panel2/tumor_2.fq

Rename units file:

```
$ mv units-example.tsv units.tsv
```

Edit configuration files

config.yaml

```
$ cp config-example.yaml config.yaml
$ vi config.yaml
```

```
samples: samples.tsv
units: units.tsv
contigs: contigs.tsv

outdir: "out"
logdir: "log"

ref:
 # Genome database of snpeff to be used in the annotation with this resource. Available databases can be checked
 with java -jar snpEff.jar databases
name: GRCh37.75
# Path to the reference genome, ideally as it is provided by the GATK bundle.
genome: /Users/patient/Desktop/hg19/hg19.fa
# Path to any database of known variants, ideally as it is provided by the GATK bundle.
known-variants: /Users/patient/Desktop/hg19/dbsnp_150.hg19.vcf.gz
```

Edit configuration files

config.yaml

```
filtering:
# Set to true in order to apply machine learning based recalibration of
# quality scores instead of hard filtering.
  vqsr: false
  hard:
  # hard filtering as outlined in GATK docs
  # (https://gatkforums.broadinstitute.org/gatk/discussion/2806/howto-apply-hard-filters-to-a-call-set)
    snvs:
    "QD < 2.0 || QUAL < 100.0 || DP < 50.0 || SOR > 3.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 ||
    ReadPosRankSum < -8.0"
    indels:
    "QD < 2.0 || QUAL < 100 || DP < 50.0 || FS > 200.0 || ReadPosRankSum < -20.0"
#depth of coverage threshold to apply to variants identified with MuTect2
depth: "DP < 30”
```

Edit configuration files

config.yaml

```
processing:
  remove-duplicates: true
  # Uncomment and point to a bed file with, e.g., captured regions if necessary,
  # see https://gatkforums.broadinstitute.org/gatk/discussion/4133/when-should-i-use-l-to-pass-in-a-list-of-intervals.
  restrict-regions: /Users/patient/Desktop/bedfiles/CCP.20131001.designed.bed
  # If regions are restricted, uncomment this to enlarge them by the given value in order to include
  # flanking areas.
  region-padding: 100
```

Edit configuration files

config.yaml

```
params:
  ...
  trimmomatic:
    pe:
      trimmer:
      # See trimmomatic manual for adding additional options, e.g. for adapter trimming.
        - "LEADING:0"
        - "TRAILING:0"
    se:
      trimmer:
      # See trimmomatic manual for adding additional options, e.g. for adapter trimming.
        - "LEADING:0"
        - “TRAILING:0"
```

Edit configuration files

config.yaml

```
resources:
  ...
  trim_reads:
      mem: 32000
      threads: 16
      walltime: 480
```

Execution

1. Activate the conda environment

```
$ conda activate snakemake
```

2. Check dry run

```
$ snakemake --use-conda -n --until merge_calls filter_mutect_2
```

```
Job stats:
job                         count    min threads    max threads
-------------------------  -------  -------------  -------------
call_variants                  24              1              1
combine_calls                  24              1              1
compose_regions                24              1              1
filter_mutect_2                 1              1              1
filter_mutect_calls             1              1              1
genotype_variants              24              1              1
hard_filter_calls               2              1              1
map_reads                       1              1              1
mark_duplicates                 1              1              1
merge_calls                     1              1              1
merge_variants                  1              1              1
mutect                          1              1              1
obtain_recal_table              1              1              1
recalibrate_base_qualities      1              1              1
samtools_index                  1              1              1
select_calls                    2              1              1
sort_bed                        1              1              1
trim_reads_pe                   1              1              1
total                         112              1              1

This was a dry-run (flag -n). The order of jobs does not reflect the order of execution.
```

Execution

3. Launch Varca

```
$ snakemake --use-conda --jobs 4 --until merge_calls filter_mutect_2
```

It takes a while, so let’s move to the other exercise while it finishes!