



PO: Precision Oncology Course

Panel Analysis - GastroIntestinal Cancer Patient



Study case

Panel 2

Tumor type: Patient with Gastrointestinal Cancer and lung metastasis

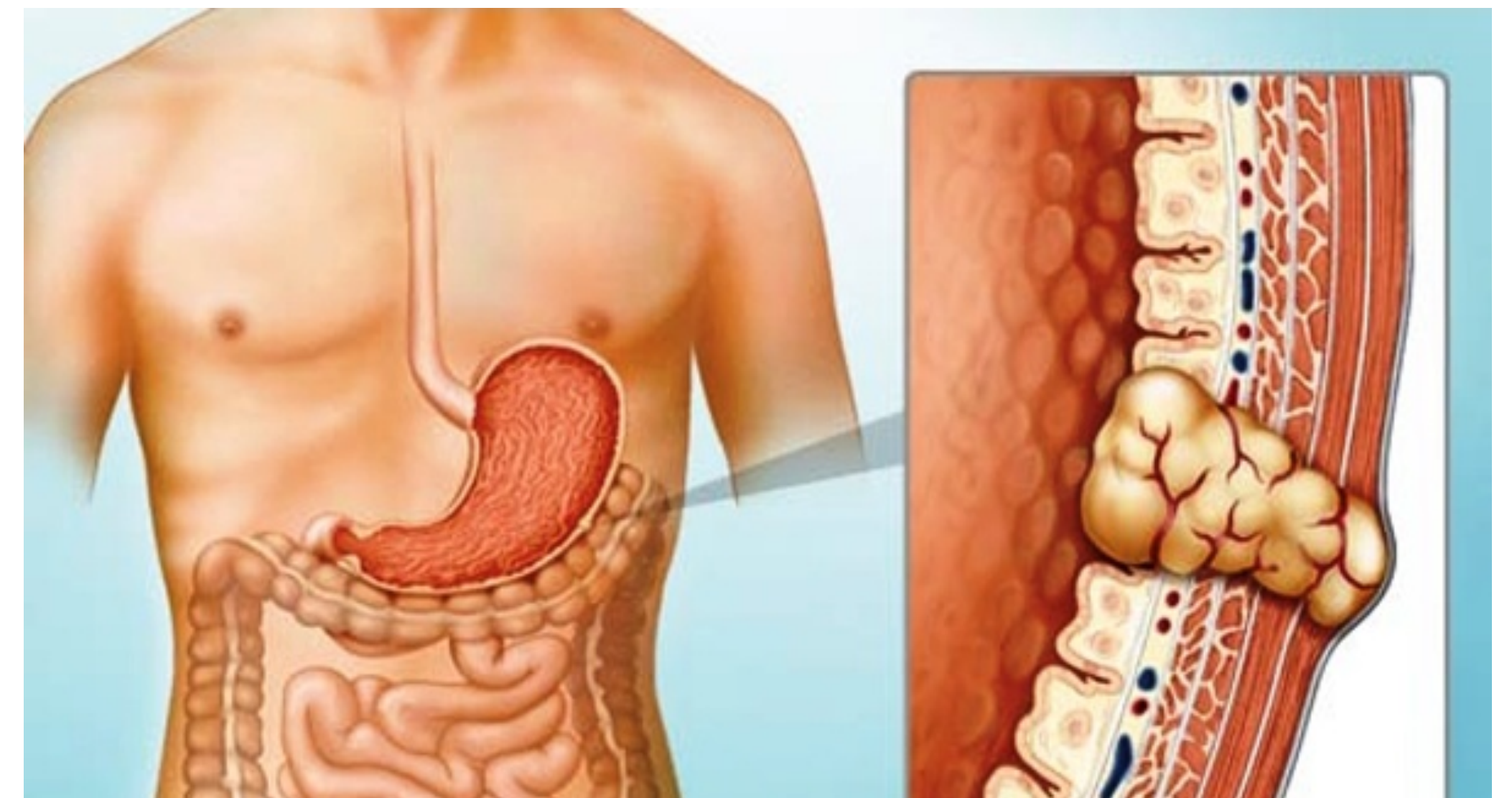
Sequencing platform: Illumina HiSeq2500

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

Samples: One tumor sample without normal tissue (paired)

Data: <https://drive.google.com/file/d/1nJtZb312a6rhUFByJ1KLBi-6avrIN47-/view?usp=sharing>

Reference genome: hg19



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

Study case

Panel 2

Tumor type: Patient with Gastrointestinal Cancer and lung metastasis

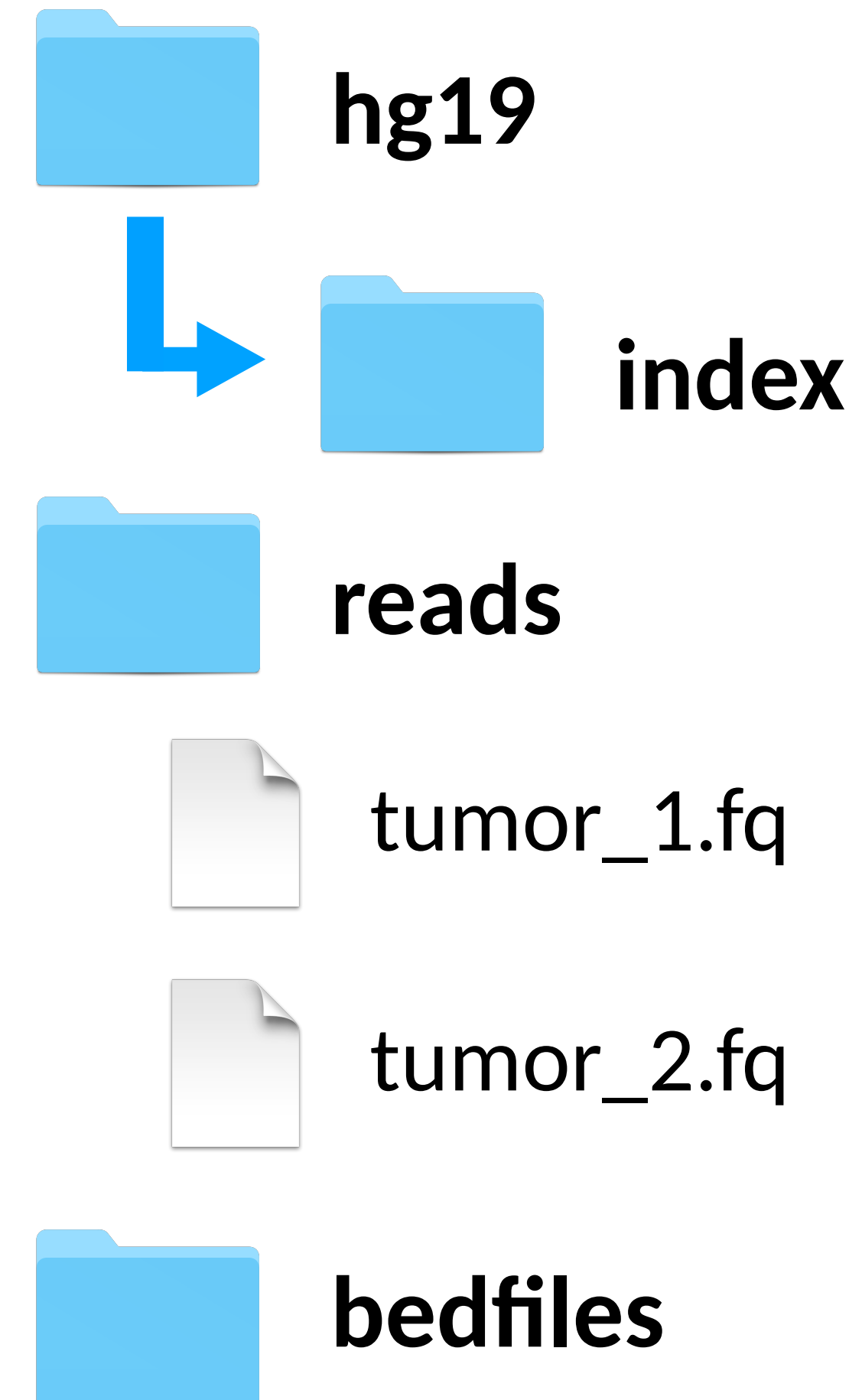
Sequencing platform: Illumina HiSeq2500

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

Samples: One tumor sample without normal tissue (paired)

Data: <https://drive.google.com/file/d/1nJtZb312a6rhUFByJ1KLBi-6avrIN47-/view?usp=sharing>

Reference genome: hg19



Steps

Configure varca

1. Move inside varca folder

```
$ cd varca
```

2. Remove previous data

```
$ rm -r previous_data
```

3. Download the data

4. Remove config files from previous analysis

```
$ rm contigs.tsv samples.tsv units.tsv config.yaml
```

Configure varca

contigs.tsv

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

```
$ touch contigs.tsv
```

Configure varca

samples.tsv

As we only have a 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.

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

group	sample	control
1	tumor	tumor

Configure varca

units.tsv

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

Rename units file: `$ cp units-example.tsv units.tsv`

sample	unit	platform	fq1	fq2
tumor	1	ILLUMINA	/path_to/panel2/reads/ tumor_1.fq	/path_to/panel2/reads/ tumor_2.fq

Configure varca

config.yaml

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


Configure varca

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: /path_to/panel2/hg19/hg19.fa
```

```
# Path to the directory with the reference indexes for the alignment.
```

```
genome_idx: /path_to/panel2/hg19/indexes/bwa_mem2
```

```
# Path to any database of known variants, ideally as it is provided by the GATK bundle.
```

```
known-variants: /path_to/panel2/hg19/dbsnp_150.hg19.vcf.gz
```

Configure varca

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"
```

Configure varca

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: /path_to/panel2/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
```



Remove the “#” and the trailing space

Configure varca

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"
```

Execution

1. Activate the conda environment

```
$ conda activate snakemake
```

2. Check dry run

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

```
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

1. Launch varca

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

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

It takes a while, so let's have lunch while it finishes!



Thanks!



cnio stop cancer