GPRO SUITE
BY BIOTECHVANA

VariantSeq

User Tutorial

Contents

1.PRELIMINARY INFORMATION

1.1 – Tutorial objective

We aim to provide end users with sufficient information and training materials to guide them through the management of server-side pipelines and workflows for characterizing Single Point Mutations and Indels (SPMI) in VariantSeq and GPRO Server Side. The tutorial will enable the user get familiarize themselves with the two execution modes VariantSeq.

Execution modes:

- **Step-by-step mode:** the different steps of the variant analysis workflow (for instance, quality analysis, preprocessing, postprocessing, mapping, variant calling, variant filtering, and annotation) are organized into individual tabs that summarize the command line interface (CLI) software available for each step.
- **Pipeline mode**: allows the user to automatically run all the steps of the workflow as a pipeline.

For more details, the manual for VariantSeq is available at https://gpro.biotechvana.com/tool/variantseq/manual/overview

1.2 – Tutorial material and case study

In this tutorial, we will use data from a case study of variant analysis published by Trilla-Fuertes et al. (2020) based on whole-exome NGS data sequenced from samples of human anal cancer. The tutorial material consists of five formalin-fixed, paraffin-embedded (FFPE) samples from patients diagnosed with localized anal squamous cell carcinoma (ASCC). These samples were analyzed by whole-exome sequencing (NextSeq500) via Illumina pair end. The sample names are provided in Table 1.

NGS Data: Five fastq files with the following SRA Accessions (**Table 1**) :

Table 1: Exome samples

SRA Accessions	**Library Name**
SRR10164002	CAN2
SRR10163991	CAN3
SRR10163980	CAN4
SRR10163969	CAN5
SRR10163960	CAN12

The 5 fastq files can be downloaded from NCBI at https://www.ncbi.nlm.nih.gov/bioproject/PRJNA573670. For questions regarding how to download this material, contact us for support in our forum at https://forum.biotechvana.com.

RefSeq material

In this tutorial, we used the Resource Bundle of GATK that is based on the Hg19 release of the human genome as a source of RefSeq. For training material, we used an interval file based on the seqCap VCRome V2 for human exome. The interval file can be downloaded by clicking seqCap_VCRome_V2_intervals_list.intervals.

For the cancer variant analysis, you need a panel of normal (PON) to filter all possible germline variants. This can be downloaded by clicking HPON.vcf and HPON.vcf.idx.

This PON was created using 11 human Iberian exome samples sequenced via Illumina technology (Illumina HiSeq 20) and Spanish populations HapMap provided by the 1000 genomes project (1000 Genomes whole exome sequencing of IBS population). The 11 samples can be downloaded from the SRA archive (http://www.ncbi.nlm.nih.gov/sra/) of NCBI with the following accessions SRR768531, SRR768530, SRR768529, SRR766062, SRR766027, SRR766011, SRR766005, SRR765982, SRR765992, SRR764760, SRR764761.

- Reference genome:

For this experiment, you need the following training material from the hg19 release:

- ucsc.hg19.dict.gz
- ucsc.hg19.fasta.fai.gz
- ucsc.hg19.fasta.gz

- Training sets and known site files:

The fastq libraries must be mapped on a reference genome (ucsc.hg19.fasta) as a RefSeq sequence . The additional files .dict and .fai are the dictionary and index files, respectively, that are associated with that sequence..

- dbsnp_138.hg19.vcf.gz
- dbsnp_138.hg19.vcf.idx.gz
- hapmap_3.3.hg19.sites.vcf.gz
- hapmap_3.3.hg19.sites.vcf.idx.gz
- 1000G_phase1.snps.high_confidence.hg19.sites.vcf.gz
- 1000G_phase1.snps.high_confidence.hg19.sites.vcf.idx.gz
- Mills_and_1000G_gold_standard.indels.hg19.sites.vcf.gz

- Mills_and_1000G_gold_standard.indels.hg19.sites.vcf.idx.gz
- af-only-gnomad.raw.sites.hg19.vcf
- af-only-gnomad.raw.sites.hg19.vcf.idx

The material (reference genome and training sets) can be downloaded from the Broad Institute FTP site at https://gatk.broadinstitute.org/hc/en-us/articles/360035890811-Resource-bundle.

The training sets and known site resources are files that include a list of variants created with machine-learning algorithms to model the properties of true variation vs. artifacts. They are required in several steps of the SPMI protocol to help the caller distinguish true variants from false positives. For more details, see https://gatk.broadinstitute.org/hc/en-us/articles/360035890831-Known-variants-Training-resources-Truth-sets

1.3 – Experiment design and support

The tutorial is designed to guide you in the steps of SPMI analysis based on a cancer exome case study using Mutect2 of GATK Mckenna (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013) and panel of normal (PON).

We use Mutec2 from GATK because this command was designed to call cancer somatic variants that are present at lower frequencies than germinal variants. For more details, see https://gatk.broadinstitute.org/hc/en-us/articles/360037593851-Mutect2. Briefly, somatic variants are acquired genomic variants that are the most common cause of cancer. They occur due to accumulated damage to the genes within an individual cell over a person's life, which is typically caused by exposure to common carcinogens such as tobacco, ultraviolet light or radiation, viruses, chemical exposures, aging, etc. Somatic variants are present at lower frequencies than germinal variants because they are not found in every cell in the body and are not passed from parent to child (see also https://voice.ons.org/news-and-views/germline-and-somatic-mutations-what-is-the-difference). The problem with cancer variants is frequencies of occurance are typically below the expected error threshold of sequencing machines. To address this issue, Mutect2 was designed to manage somatic variants that occur at low frequencies. For this reason, we also need to use a PON in the Variant Calling step. The PON is a resource used in somatic variant analysis using normal healthy samples (believed to not have any somatic alterations). Their main purpose is to capture recurrent technical artifacts in order to improve the results of the variant calling analysis. For more details, see https://gatk.broadinstitute.org/hc/en-us/articles/360035890631-Panel-of-Normals-PON-.

If you have any questions, contact us for support at https://forum.biotechvana.com. You can also visit our chatbot at https://gpro.biotechvana.com/genie.

1.4 – Installing and activating VariantSeq and the Server Side

VariantSeq is a Client Side + Server Side application. You can download the latest version of VariantSeq (the client side) at https://gpro.biotechvana.com/download/VariantSeq. To install VariantSeq on your PC, follow the instructions in the manual at https://gpro.biotechvana.com/tool/variantseq/manual. To install this app, you must have Java 11 previously installed on your PC.

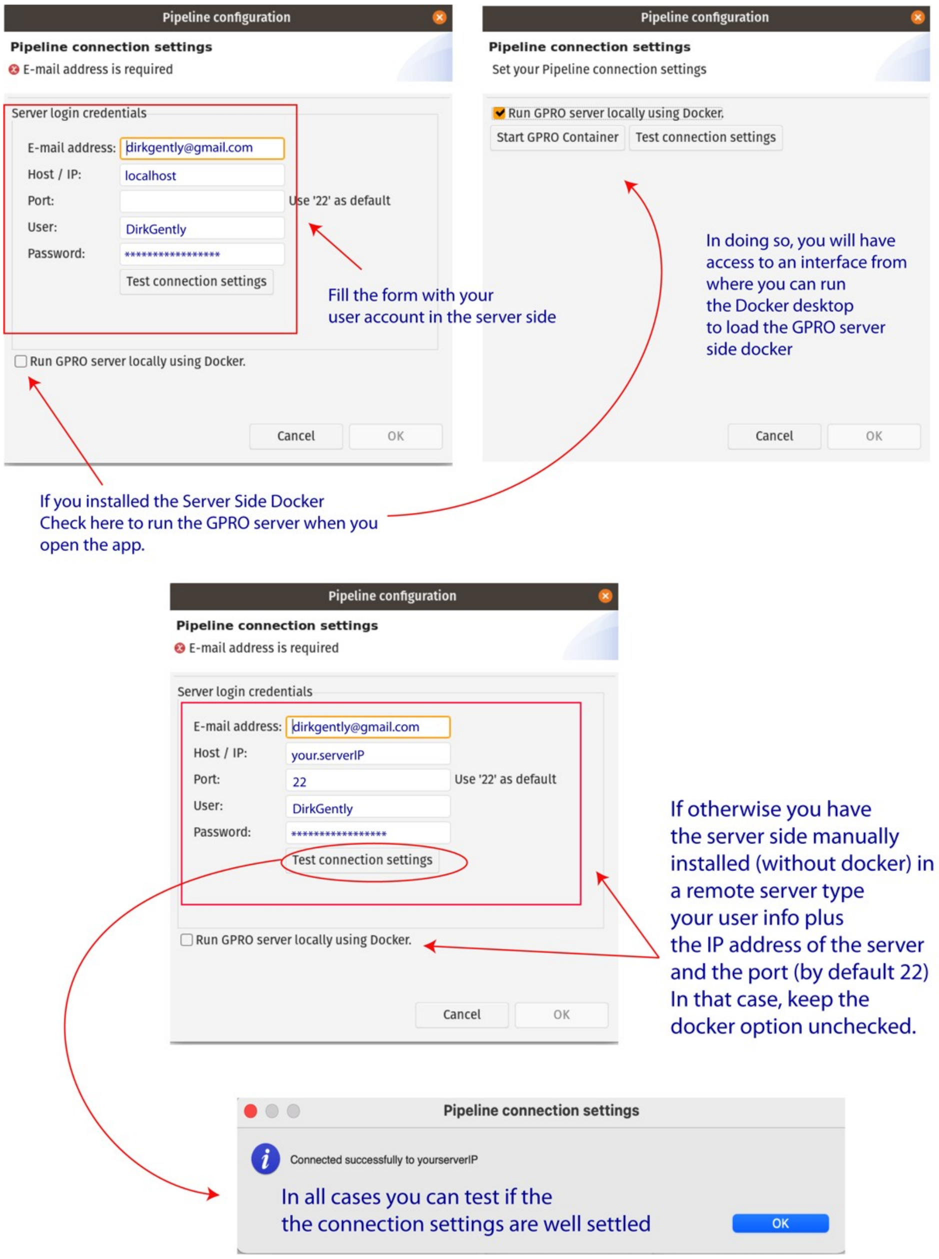
The GPRO Server Side can be installed on a PC or a remote server as a Cloud Computing resource. However, due to the complexity of installing the program,, we distribute the GPRO Server Side in a Docker container (Merkel 2014). The Docker container can be installed easily by following the steps described here: https://gpro.biotechvana.com/tool/gpro-server/manual

Once the GPRO server side docker has been installed, it must be linked to VariantSeq. To do this, go to [Preferences → Pipeline connection settings] in the top menu and type the following into the configuration Dialog (Fig.1):

1. Your email address: to receive notifications from the server.
2. Host / IP address: here you should type localhost (see Fig.1)
3. Port: This field should only be filled in case you run the server side manually. The default number will be 22.

4. Username and password: Your ID credentials provided to access the host server.

As shown in Fig.1, you can also check the option “Run GPRO server locally using Docker” to automatically start the GPRO container each time you run VariantSeq (if you have this option checked, you do not need to type in the port number). You can test if the app is connected to the Server Side by clicking on the tab “Test connection settings”. Alternatively, if you have installed the Server Side manually (without the Docker), add the IP of the remote server where the Server Side is hosted and the port information (by default 22). You should also leave the Option “Run GPRO server locally using Docker” unchecked.



Pipeline configuration

Pipeline connection settings

E-mail address is required

Server login credentials

E-mail address: dirkgently@gmail.com

Host / IP: localhost

Port: (Use '22' as default)

User: DirkGently

Password: ****************

□ Run GPRO server locally using Docker.

Fill the form with your user account in the server side

If you installed the Server Side Docker Check here to run the GPRO server when you open the app.

Pipeline configuration

Pipeline connection settings

Set your Pipeline connection settings

☑ Run GPRO server locally using Docker.

In doing so, you will have access to an interface from where you can run the Docker desktop to load the GPRO server side docker

Pipeline configuration

Pipeline connection settings

E-mail address is required

Server login credentials

E-mail address: dirkgently@gmail.com

Host / IP: your.serverIP

Port: 22 (Use '22' as default)

User: DirkGently

Password: ****************

□ Run GPRO server locally using Docker.

If otherwise you have the server side manually installed (without docker) in a remote server type your user info plus the IP address of the server and the port (by default 22) In that case, keep the docker option unchecked.

Pipeline connection settings

Connected successfully to yourserverIP

In all cases you can test if the the connection settings are well settled

Figure 1. VariantSeq connection settings and dialog

2. STEP-BY-STEP MODE TUTORIAL

In step-by-step mode, the user can individually run each step for calling and annotation of SNP and Indels from DNA and RNAseq data as a workflow created basis best practice guidelines on SPMI analysis based on the GATK (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013) and VarScan2 (Koboldt, et al. 2012), plus other tools including the Variant Effect Predictor (VEP) of Ensembl (McLaren, et al. 2016) for variant annotation. In step-by-step mode, each step of the analysis (for instance, quality analysis, preprocessing, mapping, postprocessing, variant calling, etc.) to be completed separately and in different interfaces.

2.1. – Prepare your experiment

To use the VariantSeq protocol for calling and annotating SNPs and Indels, go to: Variant Protocols → Step-by-Step Mode → SNP/Indels.

As shown in Fig.2, a new submenu appears in the workspace that lists the available options in VariantSeq for SPMI analysis:

- Preprocessing
- Mapping
- Training Sets
- Postprocessing
- Variant Calling
- Variant Filtering
- Annotation

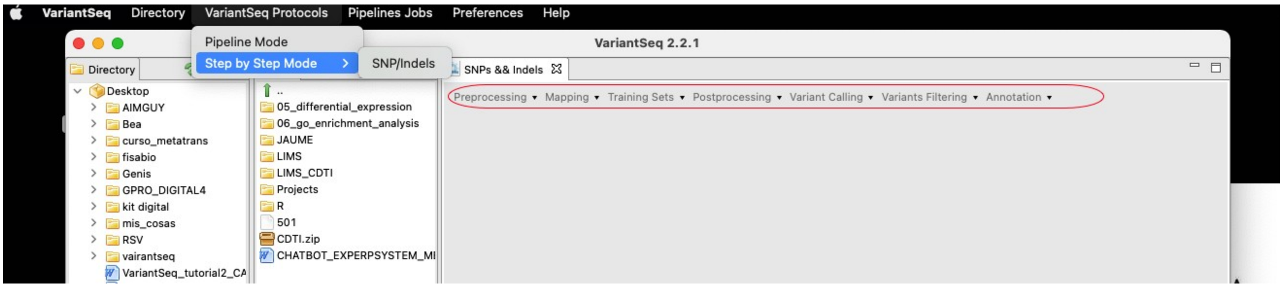


Figure 2. VariantSeq modes (SNP/Indels protocol).

As previously stated in the sections above, we call and annotate variants using distinct cancer exome samples. To do this, complete the following steps:

Quality analysis → Preprocessing → Mapping → Postprocessing → Variant Calling → Variant filtering → Variant annotation

From the Preprocessing tab, we use FASTQC (Andrews 2016) for quality analysis and PRINSEQ (Schmieder and Edwards 2011) for preprocessing.

From the Mapping tab, we use the mapper BWA (Li and Durbin 2009).

From the Postprocessing tab, we use AddReplaceGroups and MarkDuplicates from Picard Tools (Wysoker, et al. 2011) and BQSR from GATK (McKenna, et al. 2010; Cibulskis, et al. 2013).

From the Variant Calling tab, we use Mutect2 from GATK (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013).

From the Variant Filtering, we use FilterMutectCalls of GATK (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013)

From the Annotation tab, we use VEP (McLaren, et al. 2016)

If you have downloaded the tutorial material, open VariantSeq and set a directory folder to where you want aforementioned material on your PC (e.g., your desktop). The space left of the directory browser is the FTP browser for VariantSeq. This connects to the directory folder on your PC with your user account on the local host site of the server side. Right click in the FTP browser and create a folder named VariantSeq_tutorial. Next, enter this folder and create the following subfolders:

00_raw_data: to deposit the exome fastq files that you will process during the tutorial. If the fastq files are compressed, you must unzip them first.

01_quality_analysis: to deposit the results of the quality analysis.

02_preprocessed_reads: to deposit results of the preprocessing analysis.

03_refseq: to deposit for the RefSeq material needed to complete the tutorial.

04_ mapping: to deposit for the bam files produced during mapping.

05_addreplacegroups: to deposit the bam files produced during the AddReplaceGroups postprocessing analysis.

06_MarkDuplicates: to deposit the bam files produced during the Markduplicates postprocessing analysis.

07_BSQR: to deposit the bam files produced during the BSQR postprocessing analysis.

08_variant_calling: to deposit the vcf files produced during variant calling.

09_Variant_filtering: to deposit the vcf files produced during variant filtering.

10_annotation: to deposit the vcf files which contains the effects annotation for the SNPs and the Indels called and filtered.

After this, use the FTP browser to move the 5 fastq files from your directory browser on your PC to the folder 00_raw_data in your local host account on the server side. Then use the FTP browser to move the reference genome sequence, interval file, PON, and training sets (referred to in the section 1.2) from your directory browser to the folder 03_refseq created in your local host account. The process is shown in Video 1.

Video 1. Setting the directory folder and organizing the user account.

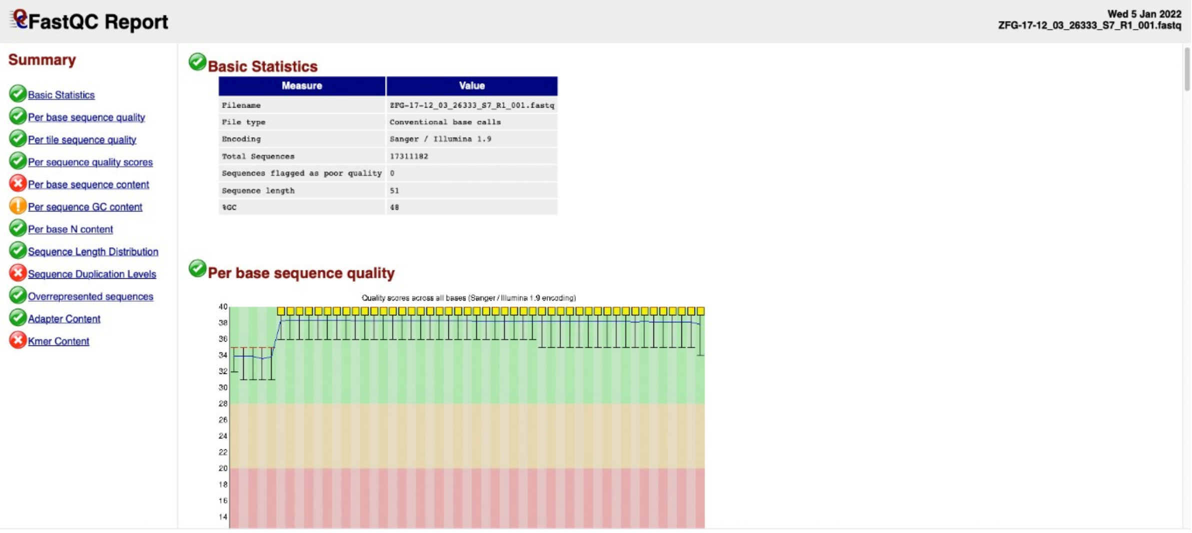
2.2 - Quality analysis and preprocessing

- Quality analysis

For quality analysis, we use FASTQC (Andrews 2016). To access the FASTQ interface in VariantSeq, go to the step-by-step menu path SNP/Indels and a submenu will appear with the following tabs: Preprocessing → Quality Analysis → FASTQC Proceed as shown in Video 2.

Video 2. Performing a quality analysis with FASTQC implementation in VariantSeq.

A FastQC report is an HTML file (Fig.3) that can be used to check the quality of a sample and it contains the sections detailed below the figure:



FastQC Report

Wed 5 Jan 2022
ZFG-17-12_03_26333_S7_R1_001.fastq

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✗ Sequence Duplication Levels
- ✓ Overrepresented sequences
- ✓ Adapter Content
- ✗ Kmer Content

✓ Basic Statistics

Measure	Value
Filename	ZFG-17-12_03_26333_S7_R1_001.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	17311182
Sequences flagged as poor quality	0
Sequence length	51
%GC	48

✓ Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14

Figure 3. FASTQC Report example.

- **Basic Statistics:** general information on the input FASTQ file such as-sample name, type of quality score encoding, total number of reads, read length, and GC content.
- **Per base sequence quality:** a box-and-whisker plot showing aggregated quality score statistics at every position along every read in the file.

- **Per tile sequence quality:** this graph will only appear if you are using an Illumina library that retains its original sequence identifiers. The graph allows you to check the quality scores from each tile across all bases to see if there were any losses in the quality within the flowcell.
- **Per sequence quality scores**: a plot of the total number of reads vs. the average quality score over the full length of that read.
- **Per base sequence content:** a plot of the percent of bases called for each of the four nucleotides at each position across all reads in the file.
- **Per sequence GC content**: a plot of the number of reads vs. GC% per read. The displayed Theoretical Distribution assumes a uniform GC content for all reads.
- **Per base N content**: percent of bases at each position or bin with no base call, i.e., 'N'.
- **Sequence Length Distribution:** a plot showing the distribution of fragment sizes in the analyzed file.
- **Sequence Duplication Levels**: percentage of reads of a specific sequence in the file that are present a given number of times in the file.
- **Overrepresented sequences**: list of sequences that appear more than expected in the file. Only the first 50bp are considered. A sequence is considered overrepresented if it accounts for ≥ 0.1% of the total reads. To identify each overrepresented sequence, it is compared to a list of common contaminants.
- **Adapter Content:** a cumulative plot of the fraction of reads where the sequence library adapter sequence is identified at the indicated base position. Only adapters specific to the library type are searched.
- **Kmer Content:** measures the count of each short nucleotide of length k (default = 7) starting at each position along the read. Any given Kmer should be evenly represented across the length of the read.

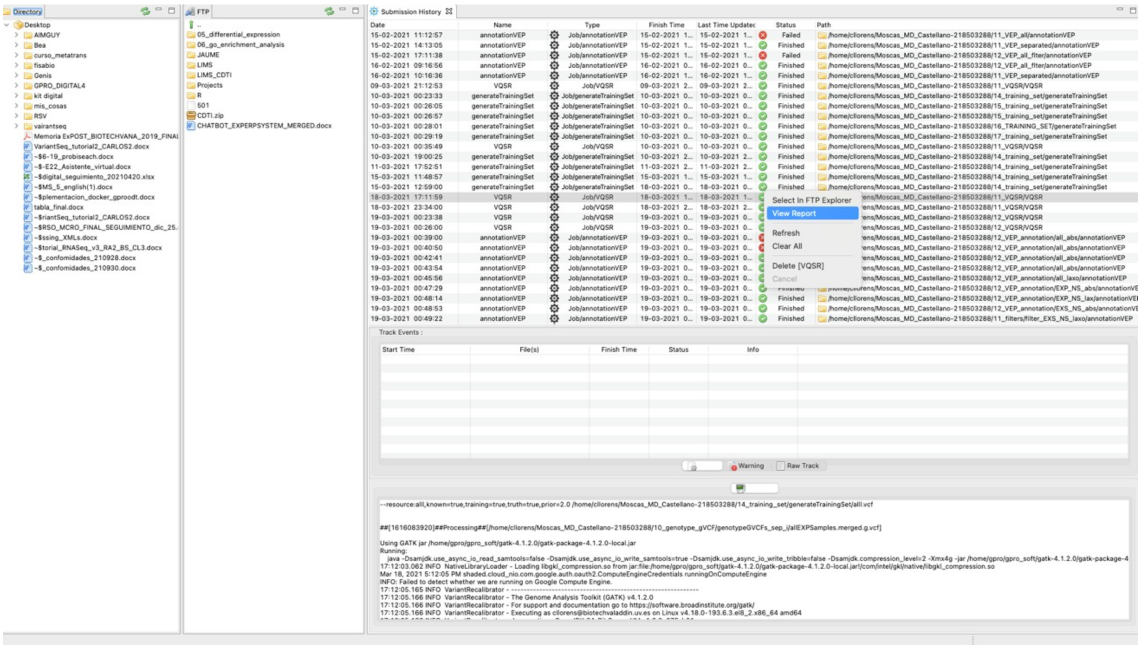
Expected results from the quality analysis

When FASTQC is finished, you will obtain a report for each fastq file in the output folder.

The expected results are available at **Quality Analysis**

For more details on the FASTQC report, visit **https://www.bioinformatics.babraham.ac.uk/projects/fastqc/**

You can see if the status of the quality analysis by going to the main menu path Pipelines Jobs → Job Tracking System. This tab gives you access to the tracking panel (Fig.4) where you can see the state of the process. By right clicking on the panel, you can update, clear, or delete a process. You can also see a log file of the process to check if something failed or which commands were used in the analysis. You can even rerun an analysis directly from the tracking panel.



Directory:

- Desktop
  - AIMGUY
  - Bea
  - curso_metatrans
  - fisabio
  - Genis
  - GPRO_DIGITAL4
  - kit digital
  - mis_cosas
  - RSV
  - vairantseq
  - Memoria ExPOST_BIOTECHVANA_2019_FINAI
  - VariantSeq_tutorial2_CARLOS2.docx
  - ~$6-19_probiseach.docx
  - ~$-E22_Asistente_virtual.docx
  - ~$digital_seguimiento_20210420.xlsx
  - ~$MS_5_english(1).docx
  - ~$plementacion_docker_gproodt.docx
  - tabla_final.docx
  - ~$riantSeq_tutorial2_CARLOS2.docx
  - ~$RSO_MCRO_FINAL_SEGUIMIENTO_dic_25.
  - ~$ssing_XMLs.docx
  - ~$torial_RNASeq_v3_RA2_BS_CL3.docx
  - ~$_confomidades_210928.docx
  - ~$_confomidades_210930.docx

FTP:

- ..
- 05_differential_expression
- 06_go_enrichment_analysis
- JAUME
- LIMS
- LIMS_CDTI
- Projects
- R
- 501
- CDTI.zip
- CHATBOT_EXPERPSYSTEM_MERGED.docx

Date	Name	Type	Finish Time	Last Time Updatec		Status	Path
15-02-2021 11:12:57	annotationVEP	Job/annotationVEP	15-02-2021 1...	15-02-2021 1...	✗	Failed	/home/cllorens/Moscas_MD_Castellano-218503288/11_VEP_all/annotationVEP
15-02-2021 14:13:05	annotationVEP	Job/annotationVEP	15-02-2021 1...	15-02-2021 1...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/11_VEP_separated/annotationVEP
15-02-2021 17:11:38	annotationVEP	Job/annotationVEP	15-02-2021 1...	15-02-2021 1...	✗	Failed	/home/cllorens/Moscas_MD_Castellano-218503288/12_VEP_all_flter/annotationVEP
16-02-2021 09:16:56	annotationVEP	Job/annotationVEP	16-02-2021 0...	16-02-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/12_VEP_all_flter/annotationVEP
16-02-2021 10:16:36	annotationVEP	Job/annotationVEP	16-02-2021 1...	16-02-2021 1...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/11_VEP_separated/annotationVEP
09-03-2021 21:12:53	VQSR	Job/VQSR	09-03-2021 2...	09-03-2021 2...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/11_VQSR/VQSR
10-03-2021 00:23:33	generateTrainingSet	Job/generateTrainingSet	10-03-2021 0...	10-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/14_training_set/generateTrainingSet
10-03-2021 00:26:05	generateTrainingSet	Job/generateTrainingSet	10-03-2021 0...	10-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/15_training_set/generateTrainingSet
10-03-2021 00:26:57	generateTrainingSet	Job/generateTrainingSet	10-03-2021 0...	10-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/15_training_set/generateTrainingSet
10-03-2021 00:28:01	generateTrainingSet	Job/generateTrainingSet	10-03-2021 0...	10-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/16_TRAINING_SET/generateTrainingSet
10-03-2021 00:29:19	generateTrainingSet	Job/generateTrainingSet	10-03-2021 0...	10-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/17_training_set/generateTrainingSet
10-03-2021 00:35:49	VQSR	Job/VQSR	10-03-2021 0...	10-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/11_VQSR/VQSR
10-03-2021 19:00:25	generateTrainingSet	Job/generateTrainingSet	10-03-2021 2...	10-03-2021 2...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/14_training_set/generateTrainingSet
11-03-2021 17:52:51	generateTrainingSet	Job/generateTrainingSet	11-03-2021 2...	11-03-2021 2...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/14_training_set/generateTrainingSet
15-03-2021 11:48:57	generateTrainingSet	Job/generateTrainingSet	15-03-2021 1...	15-03-2021 1...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/14_training_set/generateTrainingSet
15-03-2021 12:59:00	generateTrainingSet	Job/generateTrainingSet	18-03-2021 0...	18-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/14_training_set/generateTrainingSet
18-03-2021 17:11:59	VQSR	Job/VQSR	18-03-2021 1...	18-03-2021 1...	✓		ens/Moscas_MD_Castellano-218503288/11_VQSR/VQSR
18-03-2021 23:34:00	VQSR	Job/VQSR	18-03-2021 2...	18-03-2021 2...	✓		ens/Moscas_MD_Castellano-218503288/11_VQSR/VQSR
19-03-2021 00:23:38	VQSR	Job/VQSR	19-03-2021 0...	19-03-2021 0...	✓		ens/Moscas_MD_Castellano-218503288/12_VQSR/VQSR
19-03-2021 00:26:00	VQSR	Job/VQSR	19-03-2021 0...	19-03-2021 0...	✓		ens/Moscas_MD_Castellano-218503288/12_VQSR/VQSR
19-03-2021 00:39:00	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✗		ens/Moscas_MD_Castellano-218503288/12_VEP_annotation/all_abs/annotationVEP
19-03-2021 00:40:50	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✗		ens/Moscas_MD_Castellano-218503288/12_VEP_annotation/all_abs/annotationVEP
19-03-2021 00:42:41	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓		ens/Moscas_MD_Castellano-218503288/12_VEP_annotation/all_abs/annotationVEP
19-03-2021 00:43:54	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓		ens/Moscas_MD_Castellano-218503288/12_VEP_annotation/all_abs/annotationVEP
19-03-2021 00:45:56	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓		ens/Moscas_MD_Castellano-218503288/12_VEP_annotation/all_laxo/annotationVEP
19-03-2021 00:47:29	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/12_VEP_annotation/EXP_NS_abs/annotationVE
19-03-2021 00:48:14	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/12_VEP_annotation/EXP_NS_lax/annotationVE
19-03-2021 00:48:53	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/12_VEP_annotation/EXS_NS_abs/annotationVE
19-03-2021 00:49:22	annotationVEP	Job/annotationVEP	19-03-2021 0...	19-03-2021 0...	✓	Finished	/home/cllorens/Moscas_MD_Castellano-218503288/11_filters/filter_EXS_NS_laxo/annotationVEP

Start Time	File(s)	Finish Time	Status	Info

```
--resource:alll,known=true,training=true,truth=true,prior=2.0 /home/cllorens/Moscas_MD_Castellano-218503288/14_training_set/generateTrainingSet/alll.vcf


##[1616083920]##Processing##[/home/cllorens/Moscas_MD_Castellano-218503288/10_genotype_gVCF/genotypeGVCFs_sep_i/allEXPSamples.merged.g.vcf]

Using GATK jar /home/gpro/gpro_soft/gatk-4.1.2.0/gatk-package-4.1.2.0-local.jar
Running:
   java -Dsamjdk.use_async_io_read_samtools=false -Dsamjdk.use_async_io_write_samtools=true -Dsamjdk.use_async_io_write_tribble=false -Dsamjdk.compression_level=2 -Xmx4g -jar /home/gpro/gpro_soft/gatk-4.1.2.0/gatk-package-4
17:12:03.062 INFO  NativeLibraryLoader - Loading libgkl_compression.so from jar:file:/home/gpro/gpro_soft/gatk-4.1.2.0/gatk-package-4.1.2.0-local.jar!/com/intel/gkl/native/libgkl_compression.so
Mar 18, 2021 5:12:05 PM shaded.cloud_nio.com.google.auth.oauth2.ComputeEngineCredentials runningOnComputeEngine
INFO: Failed to detect whether we are running on Google Compute Engine.
17:12:05.165 INFO  VariantRecalibrator - ------------------------------------------------------------
17:12:05.166 INFO  VariantRecalibrator - The Genome Analysis Toolkit (GATK) v4.1.2.0
17:12:05.166 INFO  VariantRecalibrator - For support and documentation go to https://software.broadinstitute.org/gatk/
17:12:05.166 INFO  VariantRecalibrator - Executing as cllorens@biotechvaladdin.uv.es on Linux v4.18.0-193.6.3.el8_2.x86_64 amd64
```

Figure 4. Tracking panel to see the status of each job executed by VariantSeq on the server side.

- Preprocessing

Once the quality analysis has finished, the next step is to preprocess the fastq samples.

- Filter samples with PRINSEQ

To apply filters and clean your fastq files (SAMPLES CAN2, CAN3, CAN4, CAN5, CAN12), you can use PRINSEQ (Schmieder and Edwards 2011). You can filter all samples by quality, size, and/or Ns content. To do this, go to the step-by-step menu path SNP/Indels then Preprocessing → Trimming and Cleaning → PRINSEQ and proceed as shown in Video 3.

Video 3. Filter samples by quality, size, and Ns content with the Prinseq implementation of VariantSeq.

Optional: you can repeat the FASTQC quality analysis to check if the PRINSEQ removed all sequencing artifacts and if necessary, execute PRINSEQ a second time.

Expected results from the PRINSEQ preprocessing analysis

When PRINSEQ is complete, you will receive your fastq libraries free of adapters in your output folder.

The expected results are available in the following link **Preprocessing**

You can check if the job was successfully completed by accessing the job tracking panel of VariantSeq.

To learn more about PRINSEQ, see **http://prinseq.sourceforge.net**

NOTES: The exact methods used for preprocessing depends on personal preference: in this case, we use PRINSEQ, but you would have also to use TRIMMOMATIC (Bolger et al., 2014), CUTADAPT (Martin 2011) and other parameters.

2.3 - Mapping

The goal of mapping is to align the reads of each fastq library to the respective regions of the reference genome where the reads likely originated. Mapping the reads to the reference genome typically involves the alignment of millions of short reads to the genome using algorithms for fast alignment implemented using mapper tools.

To complete mapping with the SNP/Indel protocol, we map the preprocessed fastq files on the hg19 reference. The Step-by-Step menu offers you two DNAseq mappers; Bowtie2 (Langmead and Salzberg 2012) and BWA (Li and Durbin 2009). In this tutorial, we use BWA because is the typical mapper for exome analysis. To start, go to the Step-by-Step menu path, SNP/Indels → Mapping → DNAseq mappers→ BWA and proceed as indicated in Video 4

Video *4*. Mapping fastq libraries on the hg19 genome with the BWA implementation of VariantSeq.

Expected results from mapping analysis

When BWA is complete, you will receive a bam file per sample with the reads mapped against the reference genome.

The expected results are available at the following link **Mapping**

You can check how the job was completed by accessing the job tracking panel. Pay particular focus on the log file metrics showing the % or reads successfully mapped. An acceptable value is over 80% of reads mapped per fastq library. If the % is lower than 70%, try to preprocess the samples again to improve cleaning of the fastq libraries.

To learn more about BWA, see **http://bio-bwa.sourceforge.net**

2.4. Postprocessing

Postprocessing is needed to process the mapped reads onto the reference genome (realigning, correcting, calibrating, marking or flagging them) as according to the state-of-the-art practices. The aim of this step is to optimize alignment and minimize errors that could lead to false positives or negatives during variant calling. There are several postprocessing treatments that can be performed with VariantSeq. In this tutorial, we perform three postprocessing jobs: AddReplaceGroups, MarkDuplicates and BQSR.

- AddReplaceGroups

AddReplaceReadGroups is a command function of Picard tools (Wysoker, et al. 2011) that assigns all reads in a bam or sam file to a single new read-group (RG). This step is needed because many any tools require or assume the presence of at least one RG tag to define a "read-group" onto which each read can be assigned (as specified in the RG tag in the SAM record).

To run AddReplaceGroups, go to the Step-by-Step menu path, SNP/Indels → Postprocessing → Picard Tools → Picard–AddReplaceReadGroups and proceed as indicated in Video 5.

Video 5. Using AddReplaceGroups from Picard with VariantSeq.

Expected results from AddReplaceGroups:

-

When AddReplaceGroups is complete, you will receive a new bam file with the TAGs or labels added by AddReplaceGroup.

The expected results are available at the following link [PicardAddReplaceReadGroups]

To learn more about Picard tools and AddReplaceGroup, see

https://gatk.broadinstitute.org/hc/en-us/articles/360037226472-AddOrReplaceReadGroups-Picard- and https://broadinstitute.github.io/picard/

MarkDuplicates

MarkDuplicates is another command of Picard Tools (Wysoker, et al. 2011) to locate, mark and/or eliminate duplicated reads in a BAM or SAM file. It corrects for systematic bias by eliminating duplicated reads that arise for different reasons during a sequencing experiment (sample preparation, duplication artifacts, etc).

To run MarkDuplicates, go to the Step-by-Step menu path, SNP/Indels → Postprocessing → Picard tools → Picard–MarkDuplicates and proceed as indicated in Video 6.

Video 6. Using MarkDuplicates from Picard with VariantSeq.

Expected results from MarkDuplicates:

When the MarkDuplicates command is complete, you will receive a new bam file.

The expected results are available at the following link [MarkDuplicates]

To learn more about Picard tools and MarkDuplicates, see

https://gatk.broadinstitute.org/hc/en-us/articles/360037052812-MarkDuplicates-Picard- and https://broadinstitute.github.io/picard/

-

Base quality score recalibration (BQSR)

BQSR is a data pre-processing step performed by GATK (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013) to detect and correct systematic errors that affect the assignment of base quality scores by the sequencer. In this step, we use the training data sets described in section “1.2 - Tutorial material and case study”.

To run BQSR, go to the Step-by-Step menu path, SNP/Indels → Postprocessing → GATK Tools → BQSR and proceed as indicated in Video 7.

Video 7. Applying BQSR with the GATK implementation of VariantSeq.

Expected results from BQSR:

When the BQSR command of GATK is complete, you will receive a new bam file with this postprocessing job applied.

The expected results are available at the following link **BQSR**

To learn more about BQSR, see **https://gatk.broadinstitute.org/hc/en-us/articles/360035890531-Base-Quality-Score-Recalibration-BQSR-**

2.5 - Variant calling

VariantSeq offers provides two variant callers: GATK (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013) and VarsCan2 (Koboldt, et al. 2012). We use the GATK package and more specifically the Mutect2 tool as it is specifically designed to call somatic variants. In this step, we use the PON and one training set (dbsnp_138.hg19.vcf.gz and its index), which are described in the section “1.2 - Tutorial material and case study”.

To call somatic variants with Mutect2, go to the Step-by-Step menu path, SNP/Indels → Variant Calling → GATK based → Mutect2 and proceed as indicated in Video 8.

Video 8. Performing variant calling with the interface implemented in VariantSeq for the command Mutect2 of GATK

Expected results from variant calling:

When Mutect2 is complete,you will receive the results of the called variant in a VCF file for each bam file. VCF is a text file format (most likely stored in a compressed manner) that contains information about the called genetic variants. The VCF file presents meta-information lines (INFO, FILTER and FORMAT), a header line, and then data lines each containing information about a position in the genome for the variants called. This format also has the ability to contain genotype information for each position.

To learn more about the VCF file format, read the following document: **https://samtools.github.io/hts-specs/VCFv4.2.pdf**

The expected results are available at the following link **Mutect2**

To learn more about Mutect2, see **https://gatk.broadinstitute.org/hc/en-us/articles/360037593851-Mutect2**

2.6 - Variant filtering

Variant filtering identifies confident variants called by Mutect2 and removes those that do not pass the filtering threshold to eliminate potential false positives. VariantSeq provides use of several GATK tools for variant filtering (for more details, see the VariantSeq manual at https://gpro.biotechvana.com/tool/variantseq/manual). As we used Mutect2 to call the variants, we use two GATK tools for filtering: “Cross-sample Contamination” to generate contamination tables and “FilterMutectCalls to filter using the contamination tables generated by CalculateContamination.

Cross-SampleContamination calls a small pipeline to generate the contamination tables based on two GATK tools: GetPilupSummaries and CalculateContamination (McKenna, et al. 2010; DePristo, et al. 2011; Cibulskis, et al. 2013). To perform this analysis, you will need the bam files from last analysis in postprocessing step as well as the VCF files used in the variant calling step. As you do not have normal pairs for your tumor samples, you need to select Tumor only mode in the input field to upload only bam from tumor samples. To start, go to the Step-by-Step menu path, SNP/Indels → Variants Filtering → Cross-Sample Contamination and proceed as shown in Video 9.

Once you have the contamination tables, you can perform the next analysis where we will use the vcf files and the contamination sample to perform the filtering with FilterMutectCalls. To start, go to the Step-by-Step menu path, SNP/Indels → Variants Filtering → FilterMutectCalls and proceed as shown in Video 9.

Video 9.- Filtering Variants called with Mutect2 in two steps: one using Cross-SampleContamination that applies a pipeline based on GetPilupSummaries and CalculateContamination to generate contamination tables and the second job based on a interface implementation of FilterMutectCalls that performs the filters.

Expected results from variant filtering:

This analysis is performed in two steps:

- When Cross-SampleContamination is complete, with you will receive a contamination table for each VCF file.

- When FilterMutectCalls is complete, you will receive a new VCF file for each sample with the filter applied.

The expected results are available at the following link **Cross-sample contamination/FilterMutectCalls**

For more info about GetPilupSummaries, see **https://gatk.broadinstitute.org/hc/en-us/articles/360037593451-GetPileupSummaries**

For more info about CalculateContamination, see **https://gatk.broadinstitute.org/hc/en-us/articles/360036888972-CalculateContamination**

For more info about FilterMutectCalls, see **https://gatk.broadinstitute.org/hc/en-us/articles/360036856831-FilterMutectCalls**

2.7 – Annotation

The annotation step consists of annotating the functional effects to the called variants. VariantSeq uses the Variant Effect Predictor (VEP) of Ensembl (McLaren, et al. 2016) as a toolset for the analysis, annotation and prioritization of genomic variants in coding and non-coding regions. To start, go to the Step-by-Step menu path, SNP/Indels → Annotation → VEP-Variant Effect Predictor and proceed as shown in Video 10.

Video 10.- Annotating your called variants with the VariantSeq implementation of VEP

Expected results from Annotation:

When VEP is complete, you will receive a report in html format (web based) for each VFC file. This file can be opened with any internet browser or a new VCF file with the extension .txt including new annotations regarding the consequences of the variants.

To learn more about the VCF file format, read the following document **https://samtools.github.io/hts-specs/VCFv4.2.pdf**

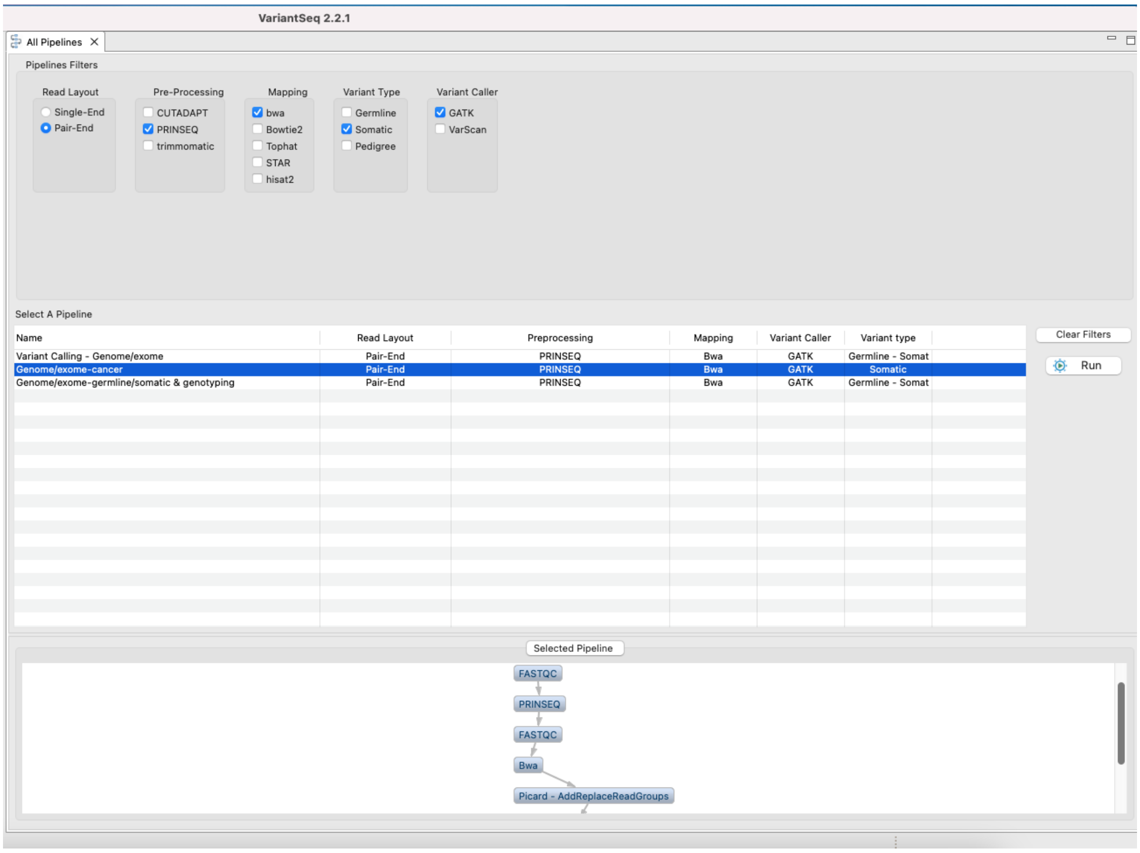
The expected results are available at the following link **Variant Effect Predictor (VEP)**

To learn more about VEP, see **https://www.ensembl.org/info/docs/tools/vep/index.html**

3. PIPELINE MODE TUTORIAL

The pipeline mode of VariantSeq allows you to execute all steps of a protocol automatically as a pipeline. To perform the tutorial in pipeline mode, click on the “VariantSeq protocols” tab in the Top menu and select the option Pipeline Mode. A list will appear to show the available pipelines and filters based on types of reads, protocol mode, etc (Fig. 5). You only need to apply the filters and click run. This will pass you to the next section, which is a set of nested interfaces to:

1. Upload the input files and Refseq needed by the pipeline (i.e., fastq files, reference genome, PON and Training, and truth sets).
2. Declare the output folder to deposit the output results.
3. Declare the experiment design (samples to analyze etc).
4. Configure the parameters and options for each tool used at each step (for example, “FastQC” for quality analysis, “PRINSEQ” for preprocessing, “BWA” for mapping, “Picard Tools and GATK” for postprocessing, and GATK for calling and filtering and VEP for annotation).
5. Run the configured pipeline.



VariantSeq 2.2.1

All Pipelines

Pipelines Filters

Read Layout: ○ Single-End ● Pair-End

Pre-Processing: □ CUTADAPT ☑ PRINSEQ □ trimmomatic

Mapping: ☑ bwa □ Bowtie2 □ Tophat □ STAR □ hisat2

Variant Type: □ Germline ☑ Somatic □ Pedigree

Variant Caller: ☑ GATK □ VarScan

Select A Pipeline

Name	Read Layout	Preprocessing	Mapping	Variant Caller	Variant type
Variant Calling - Genome/exome	Pair-End	PRINSEQ	Bwa	GATK	Germline - Somat
Genome/exome-cancer	Pair-End	PRINSEQ	Bwa	GATK	Somatic
Genome/exome-germline/somatic & genotyping	Pair-End	PRINSEQ	Bwa	GATK	Germline - Somat

Figure 5. Pipeline configuration interface.

When you have the tutorial material ready, access the pipeline designer interface VariantSeq Protocols → Pipeline mode and proceed as shown in video 11.

Video 5.- Protocol for SPMI analysis with VariantSeq using the pipeline mode

Expected results from SPMI protocol using the Pipeline execution mode:

When the pipeline mode is complete, you will receive the differential expression results with the reads mapped against the reference genome.

The expected results are available at the following link **Pipeline mode results**

You can check if the job was successfully completed by accessing the job tracking panel of VariantSeq

To learn more about any tool used in the pipeline and their outputs, see their respective manuals as indicated in previous sections of this tutorial.

4. BIBLIOGRAPHY

- Andrews S. 2016. FastQC: a quality control tool for high throughput sequence data. Bioinformatics 2015;31(2):166-169.

- Cibulskis K, Lawrence MS, Carter SL, Sivachenko A, Jaffe D, Sougnez C, Gabriel S, Meyerson M, Lander ES, Getz G. 2013. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. Nat Biotechnol 31:213-219.

- DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, Rivas MA, Hanna M, et al. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet 43:491-498.

- Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, Miller CA, Mardis ER, Ding L, Wilson RK. 2012. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res 22:568-576.

- Langmead B, Salzberg SL. 2012. Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357-359.

- Li H, Durbin R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25:1754-1760.

- McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 20:1297-1303.

- McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GRS, Thormann A, Flicek P, Cunningham F. 2016. The Ensembl Variant Effect Predictor. Genome Biology 17:122.

- Merkel D. 2014. Docker: lightweight Linux containers for consistent development and deployment. Linux Journal 2014:2.

- Schmieder R, Edwards R. 2011. Quality control and preprocessing of metagenomic datasets. Bioinformatics 27:863-864.

- Trilla-Fuertes L, Ghanem I, Maurel J, L GP, Mendiola M, Pena C, Lopez-Vacas R, Prado-Vazquez G, Lopez-Camacho E, Zapater-Moros A, et al. 2020. Comprehensive Characterization of the Mutational Landscape in Localized Anal Squamous Cell Carcinoma. Translational oncology 13:100778.

- Wysoker A, Tibbetts K, Fennell T. 2011. PicardTools 1.5.3.