Identification of somatic and germline variants from tumor and normal sample pairs - Reproduced by Aanuoluwa Adekoya (Team Crick)

Introduction

Mutation, an alteration in the arrangement of nucleic acid sequence of an organism's genome, leads to the changing of the structure of a gene, resulting in a variant form. These mutations can be germline, somatic, or loss of heterozygosity (LOH). For germline mutation, the variant occurs in the sperm or egg cell and is passed directly from a parent to an embryo (child) at the time of conception. The variant is copied into every cell as the embryo divides and develops. Somatic mutation is acquired after birth, and the variants are not found in every cell in the body. For LOH, one normal copy of a gene or group of genes is lost**.** If pathogenic or occurring in essential genes, these mutations can lead to the onset of many diseases such as cancer (mutation in tumor suppression genes). The identification of variants helps in disease diagnosis and treatment. For germline variations, comparing the sample sequence to the reference genome sequence provides information about the variants present. However, for somatic variations, extra work is required. Cancer cells, for example, have more genetic changes than normal cells, and it is individual-specific as these changes continue to occur through the cancer developmental phase. For this reason, the identification of somatic variants requires that normal and diseased tissue is obtained from the same patient. The sequences from these can then be compared to the reference genome.

Aim: This task aims to reproduce a workflow that identifies germline and somatic variants and variants affected by LOH using both healthy and tumor tissue. Our results would be used to report variant sites, and genes affected that could likely be the cause of the disease.

Importance: The identification of these variants can be useful in the prevention and early detection of cancers and other genetic diseases. Knowledge of these different types of variants can be used to improve precision medicine, manipulate therapeutic strategies, and develop novel therapeutics for the management.

Methodology

Data collection

Healthy and tumor reads from chromosomes 5,12, and 17 from a cancer patient alongside the reference genome were retrieved from Zenodo (zenodo.org), a free open-access general-purpose repository developed under the European open AIRE program and operated by CERN. Variant annotation datasets and gene-level annotation files were also retrieved from this database.

Pipelines

Linux and Galaxy (usegalaxy.eu) were used.

Linux: This pipeline was used for data collection. The wget command was used to get the files and dataset from the website using their URLs. The reference genome file was a .gz file, and the gunzip command was used to unzip it, as shown below.

```
#downloading sample dataset and reference genome
  wget https://zenodo.org/record/2582555/files/SLGFSK-N_231335_r1_chr5_12_17.fastq.gz
  wget https://zenodo.org/record/2582555/files/SLGFSK-N_231335_r2_chr5_12_17.fastq.gz
  wget https://zenodo.org/record/2582555/files/SLGFSK-T_231336_r1_chr5_12_17.fastq.gz
  wget https://zenodo.org/record/2582555/files/SLGFSK-T_231336_r2_chr5_12_17.fastq.gz
  wget https://zenodo.org/record/2582555/files/hg19.chr5_12_17.fa.gz
  gunzip hg19.chr5_12_17.fa.gz
#renaming sample datasets for easy access
  mv SLGFSK-T_231336_r1_chr5_12_17.fastq.gz Tumor_R1.fastq.gz
  mv SLGFSK-T_231336_r2_chr5_12_17.fastq.gz Tumor_R2.fastq.gz
  mv SLGFSK-N_231335_r1_chr5_12_17.fastq.gz Normal_R1.fastq.gz
  mv SLGFSK-N_231335_r2_chr5_12_17.fastq.gz Normal_R2.fastq.gz
#downloading variant annotations datasets
 wget https://zenodo.org/record/2581873/files/hotspots.bed
 wget https://zenodo.org/record/2581873/files/cgi_variant_positions.bed
 wget https://zenodo.org/record/2581873/files/01-Feb-2019-CIVic.bed
 wget https://zenodo.org/record/2582555/files/dbsnp.b147.chr5_12_17.vcf.gz
#downloading gene-level annotation files
 wget https://zenodo.org/record/2581881/files/Uniprot_Cancer_Genes.13Feb2019.txt
 wget https://zenodo.org/record/2581881/files/cgi_genes.txt
 wget https://zenodo.org/record/2581881/files/01-Feb-2019-GeneSummaries.tsv
```

Galaxy: Galaxy was used to analyze the datasets. A new workflow and history were created and named 'hackbio stage two.' The datasets were uploaded to galaxy using the 'choose local files' option. The data types were set: fastasanger.gz for the samples, fasta for the reference genome, vcf for the dbsnp dataset, bed for the other variant annotation datasets, and tabular for the gene-level annotation files. The following tools were used for analysis: FastQC, MultiQC, trimmomatic, map with BWA-MEM, Filter bam on a variety of attributes, RmDup, BamLeftAlign, CalMD, VarScan Somatic, SnpEff eff, GEMINI load, GEMINI annotate, GEMINI query, Join two files and Column arrange by header name. These tools were used in the order in which they were written.

Results and Discussion

Quality Control

Before analyzing or manipulating any high throughput sequencing reads, checking for the quality of the reads is important as this provides insights into the integrity of the reads and the amount of contamination present if any. For this, the FastQC tool was used to check the quality of the 4 sample reads, which generated four outputs and were merged into one summarised report using the MultiQC tool. The summary showed that the reads had the same sequence length of 101bp, but the diseased tissue (tumor) had unusual per sequence GC count and more duplicated reads; that is, if the reads are deduplicated, only less than 60% of the reads would be left (Table 1, Fig. 1).

Table 1: Summary of MultiQC results of the four sample datasets

Sample Name	% Dups	% GC	M Seqs
Normal_R1_fastq_gz	26.4%	49%	10.6
Normal_R2_fastq_gz	25.3%	49%	10.6
Tumor_R1_fastq_gz	43.0%	53%	16.3
Tumor_R2_fastq_gz	41.9%	53%	16.3

Dups = Duplicate Reads
GC = GC content
M Seqs = Total Sequences in millions

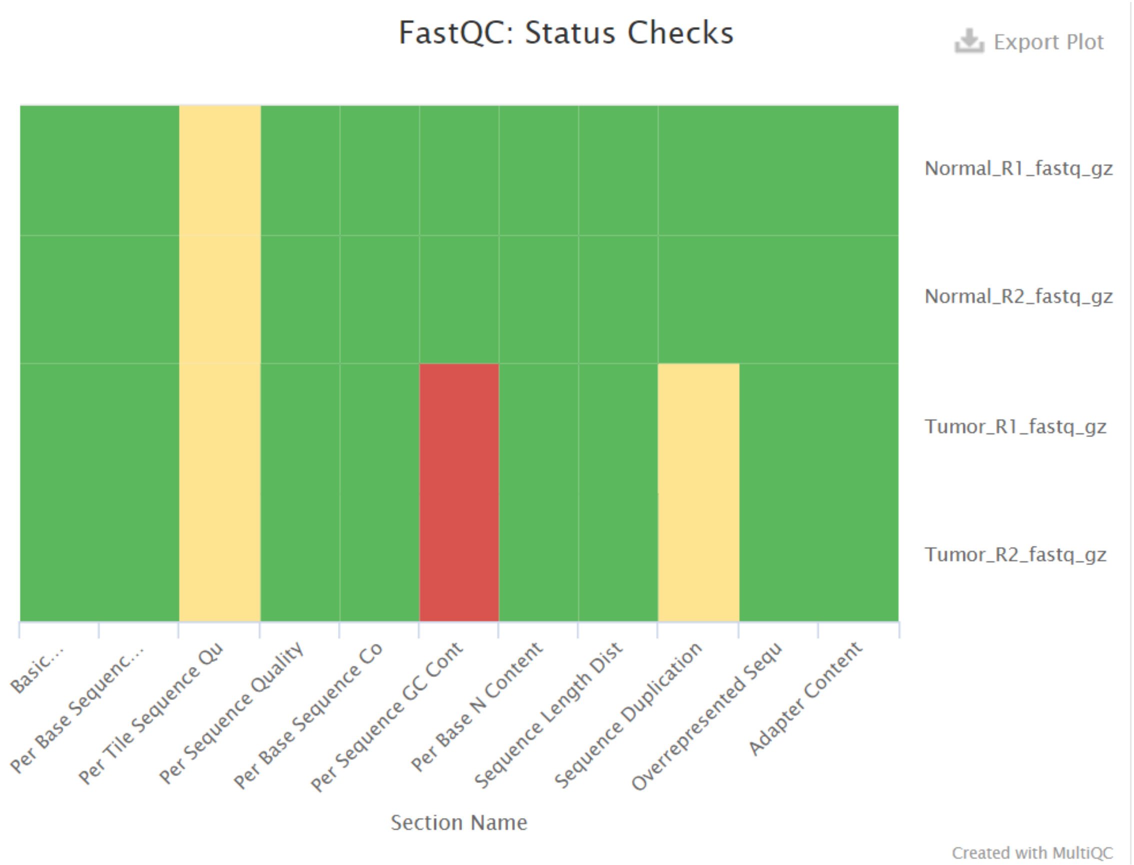


Figure 1: Summary of FastQC report of reads generated by MultiQC. Green means entirely normal, yellow means slightly abnormal, and red means very unusual. R1=forward read and R2 = reverse read for paired-end samples.

The overall report showed that the quality of the reads was good however, a few adapters were observed (Fig. 2), and we used the trimmomatic tool to trim these since we know that adapter contamination will interfere with our mapping of reads to the genome and other downstream processes. To use trimmomatic, we specified that the tool should perform initial ILLUMINACLIP to remove the adapters and we used the Truseq3 paired end adapter for MiSeq and HiSeq. We also stated the following conditions be considered: Maximum mismatch count = 2, Accuracy of the match between the two ‘adapter ligated’ reads for PE palindrome read alignment =30, Accuracy of the match between any adapter against a read =10, Minimum length of adapter that needs to be detected (PE specific/ palindrome mode = 8 and always keep both reads. Since trimmomatic could remove low quality reads, during adapter trimming, we also used some other operations of trimmomatic to remove low quality reads using the following conditions: HEADCROP:3, that is remove 3 bases from the start of the read; TRAILING:10, that is the minimum threshold quality required to keep a base is 10 and MINLEN:25, that is drop reads below a minimum of 25 bases. After trimming, we ran FastQC and MultiQC again to check the quality of the trimmed reads. The trimming process reduced the adapters as no sample was found with any adapter contamination > 0.1%. However, the process reduced the sequence length distribution quality as some reads were lost (Fig. 3).

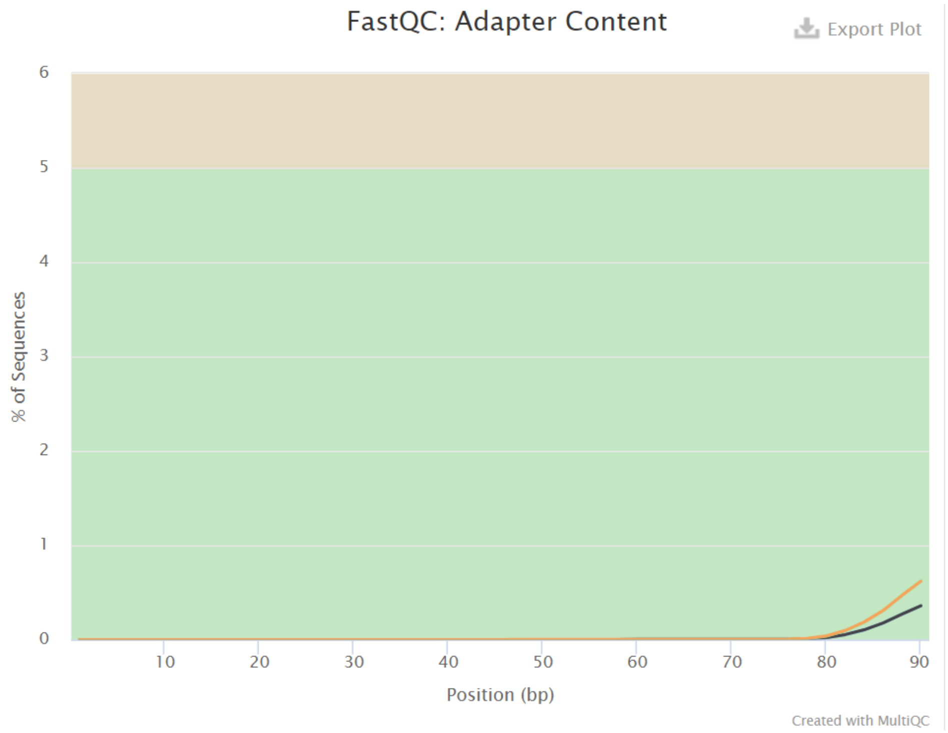


Fig 2. Adapter content of the sample reads prior to trimming. Black = normal sample. Orange = tumor sample.

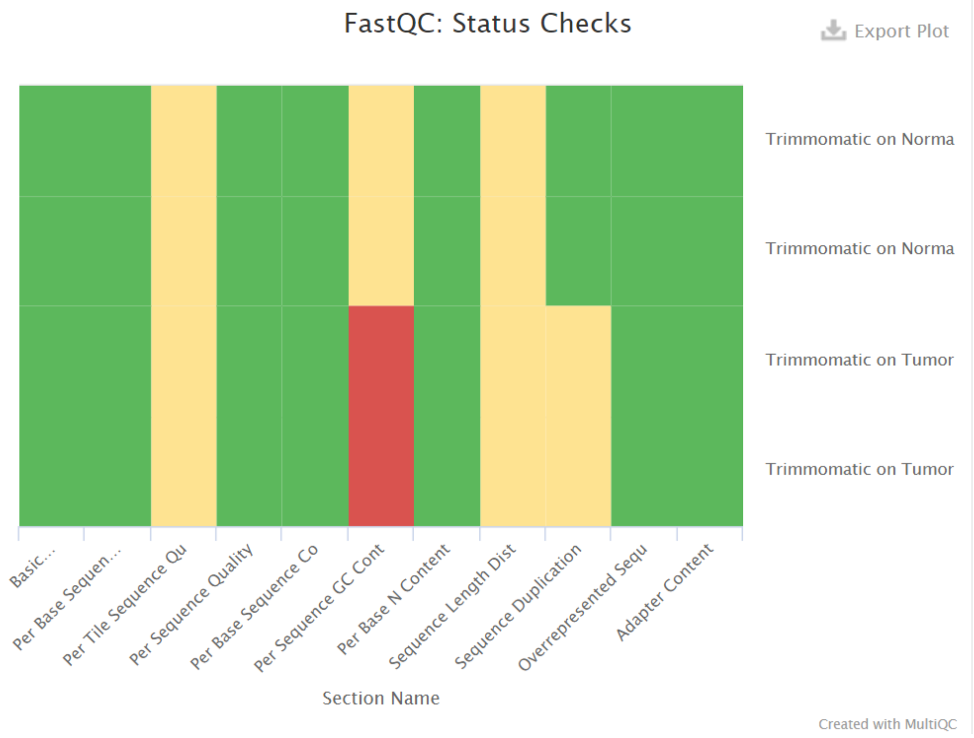


Figure 3: Summary of FastQC report of the trimmed reads generated by MultiQC. Green means entirely normal, yellow means slightly abnormal, and red means very unusual. R1 and R2 = paired-end samples.

Mapping and Mapped Reads Postprocessing

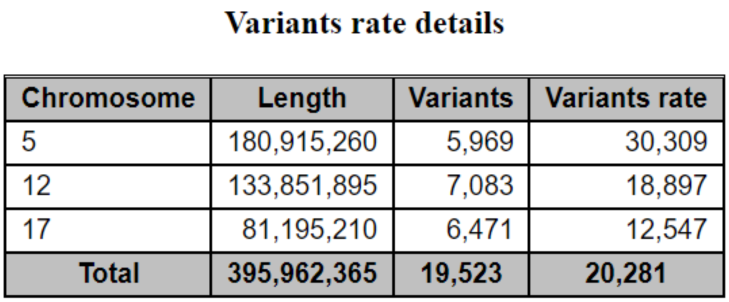
Post quality control, the trimmed reads (Normal R1 and 2; Tumor R1 And 2) were mapped to the imported human genome as the reference genome using BWA-MEM and we chose the read group identifier and sample name based on the information provided in the dataset name. Mapping was done to assign the reads to their position or location on the genome so that we can know where the sequencing reads came from. This generated a single output for the normal reads and one output for the tumor reads. However, since mapping is a probabilistic function, we had to filter the bam file that was generated. The Filter BAM datasets on a variety of attributes tool was used and MapQuailty, isMapped and isMateMapped were the specified filters. Mapquality is the probability that a read is aligned the wrong place and a value of 0 means that the read maps to multiple locations. So we specified that the mapquality be >=1 and we specified that isMapped and isMateMapped be yes because we want our output to be reads that were successfully mapped to a reference with their pairs. Since the sequencing process involved PCR and PCR products, RmDup was used to remove the PCR duplicates from our filtered BAM file. On start positions where multiple reads were aligned to the same exact position, RmDup retained read pairs with the highest mapping quality and discarded the other pairs. We specified that BAM is paired-end and selected ‘no’ for ‘treat as single end’. These files contain insertions and deletions which could be the same variant at different positions and we do not want these to affect downstream analyses, therefore we used BAMLeftAlign tool to merge and place all the indels (insertions deletions) at the left most position. The built-in Human:hg19 genome was used as the reference genome for this action. During alignment or mapping, there is a probability that a read was incorrectly mapped and the mapping quality score indicates this, hence we used the CalMD to recalibrate the read mapping qualities of the left-aligned dataset from normal and tumor tissue. We used the inbuilt hg19 genome as the reference again and for additional options, we chose advanced options so we selected 50 as our coefficient to cap the quality of poorly mapped reads as studies have shown that for variant calling, reads mapped with BWA-MEM be capped at this rate. For this process, we selected no for the calculation of BAQ scores and the prompt to change identical bases. Again, we had two outputs, one for normal and the other for the tumor dataset. The last step in our mapped reads postprocessing was refiltering. The previous process used CalMD which could have altered some read mapping quality scores and set it to 255. Unfortunately, a read mapping quality score of 255 indicates unavailable or undefined mapping quality which is not exactly good. So we used the Filter BAM datasets on a variety of attributes tool again to eliminate reads with undefined mapping quality and our filter condition was just mapQuality with the score being set to less than or equal to 254 (<=254). This was the end of our mapped reads postprocessing step and we had out filtered normal and tumor reads as the output.

Variant Calling and Classification

Variant calling is the process by which we identify variants (mutations) from sequence data and classify them as germline, somatic or loss of heterozygosity event variants. The essence of the next step was to detect variant alleles in tumor or normal sample pairs in our normal and tumor in our sample data, call sample genotypes at variant sites and classify them. The final filtered reads with high quality that were generated from our last mapped reads postprocessing step were used for this purpose and the VarScan Somatic tool was used. The imported hg19 was used as the reference genome, and for the first time both the normal and tumor reads were used as input at the same time. Input one was the filtered normal read while input two was the filtered tumor read. The estimated purity content for the normal read was set to 1 while it was set to 0.5 for the tumor read. Customized setting was used for variant calling and minimum base quality wass set to 8 while minimum mapping quality was set to 1. These two parameters were necessary because the MultiQC report showed that our reads are of good quality and the base quality can be increased without losing a significant portion of our data. The minimum mapping quality was set to 1 because, again, CalMD might have set the mapping quality of some reads to zero (mapping to multiple locations) and we need these to be filtered out. Default values were used for posterior variant filtering. Note that this step took both the tumor and normal sample reads and had a single output, a VCF file.

Variant Annotation Reporting

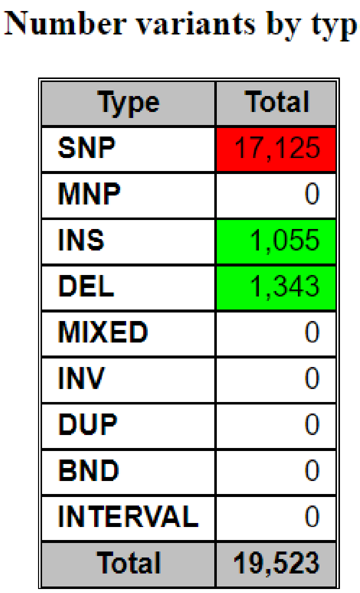
Variant annotation is the process of assigning information to DNA variants. Annotation results can have a strong influence on the ultimate conclusions of disease studies hence there is need for a correct annotation process. Functional, genetic and clinical based annotations were added to our called variants. First functional annotation was added using the SnpEff eff tool. SnpEff eff annotates and predicts the effects of genetic variants on genes and proteins. The VarScan output in the variant calling phase was the input file while the locally installed homo sapiens: hg19 was the reference genome. The input format and output formats were VCF. Two outputs were produced, the dataset and the HTML stat. Figure 4 and 5 below show the summary of the variant rate and type. The variants had a Missense / Silent ratio of 0.7864 i.e majority of the mutations were silent variants with no deleterious effect at the amino acid level.



Variants rate details

Chromosome	Length	Variants	Variants rate
5	180,915,260	5,969	30,309
12	133,851,895	7,083	18,897
17	81,195,210	6,471	12,547
Total	395,962,365	19,523	20,281

Fig. 4 Variant rate details after running SnpEff eff for functional annotation



Number variants by type

Type	Total
SNP	17,125
MNP	0
INS	1,055
DEL	1,343
MIXED	0
INV	0
DUP	0
BND	0
INTERVAL	0
Total	19,523

Fig. 5 Variant type after running SnpEff eff for functional annotation

The functional annotation showed the type of mutations that the 19,523 variants are. In addition to the type and function seen, we wanted to see if these mutations have been seen in the human population and if they have been associated with any disease. Should they have been seen, we wanted to see the prevalence. To do this, we loaded our functional annotation output into the GEMINI database using the GEMINI load tool. The GEMINI database would allow us to carry out more annotations and it will also add some annotations such as prevalence to our variants. We also loaded the GERP scores (a score used to calculate the conservation of each nucleotide in multi-species alignment), CADD scores (the deleteriousness of single nucleotide variants as well as insertion/deletions variants in the human genome), Sample genotypes and variant INFO field into the database. In addition to these ‘already made’ annotations, we used the GEMINI annotate tool to extract some more information about our variants that have been loaded to the GEMINI database. VarScan somatic calculated three important values for the variants (see VarScan Somatic report), however, GEMINI load could not recognize these. Hence, it is important that we add them. These are somatic status (SS), Germline p-value (GPV), Somatic p-value (SPV). These values are in the INFO column of the VarScan report. Somatic Status (SS), the first, is an integer in which 0 = Reference, 1 = Germline variant, 2 = Somatic variant, 3 = LOH and 5 = Unknown. Germline p-value (GPV) for variants with somatic status of one, i.e germline variants, is the error probability associated with the status call, hence it is a float. Somatic p-value (SPV), this is the error probability associated with status calls of variants with somatic status of 2 and 3 and it is also a float. Here we selected that the first annotation found be returned for the three values. We added these values using the GEMINI annotate tool and the input file was the GEMINI load output/file from our initial step of loading our functional annotation output into GEMINI.

After we had added these INFO to the database, we then added additional annotations apart from the GEMINI and VarScan Somatic obtained ones. Here we will use variants annotations retrieved from Cancer Hotspots and dbSNP and variant and gene information retrieved from Cancer and Biomarkers database of the Cancer Genome Interpreter (CGI) project and CIViC database (note that these data were downloaded through Zenodo). We repeatedly used the GEMINI annotate tool for these four steps and the output of each preceding one was the input for the next. For dbSNP, while creating the GEMINI database, it already checked if the variants occurred in dbSNP and stored their IDs but we needed to extract the dbSNP SNP Allele Origin (SAO) and add it as "rs_ss" column to the existing database. The last output from GEMINI annotate was the input and the dbsnp.vcf file was the annotation source. We extracted the data as integers and stored only the first annotation found. For Cancer Hotspots, the last ouput generated from annotating dbSNP was the input, the imported hotspot.bed file was annotation source to extract "q-values" of overlapping cancer hotspots and add them as "hs_qvalue" column to the existing database.These values are numbers with decimal precision and the smallest numeric values were stored. For CIViC, the output of the last Gemini annotate for cancer hotspots was the input and the imported CIViC.bed file was the annotation source. We extracted the fourth element from this file and added them as a list of "overlapping_civic_urls" to the existing Gemini database. This means that the hyperlinks of the overlapping CIViC sites were in the fourth column and this made our choice of data type to be text. For CGI, the last output was the input and the imported CGI.bed file was annotation source. Here we wanted to extract a binary indicator which is a boolean to let us know if our variant had any match in the annotation source where 1 will mean yes, match found and 0 will mean no, match not found. The information extracted was recorded in the Gemini database as "in_cgidb" being used as the column name.

Post annotation, we wanted to report subsets of variants of choice and these choices can be specified using the GEMINI query tool. Here we decided to obtain bonafide somatic variants and we exploited the somatic status info provided by our VarScan Somatic output to achieve this by setting somatic status to 2. We used the fully annotated variants (CGI GEMINI annotate output) as our input and basic variant query constructor as Build GEMINI query. For "Insert Genotype filter expression" we used gt_alt_freqs.NORMAL <= 0.05 AND gt_alt_freqs.TUMOR >= 0.10, that is, read only variants that are supported by less than 5% of the normal sample, but more than 10% of the tumor sample reads. For Additional constraints expressed in SQL syntax, we set somatic_status = 2 (See VarScan Somatic Output for verification). By GEMINI default, the report of this run would be output in tabular format; and a column header is added to it output. Since we had the choice to select our headers, the following columns were selected : chrom; start; ref; alt. For ‘additional columns (comma-separated)”, gene, aa_change, rs_ids, hs_qvalue, cosmic_ids. These columns are obtained from the variants table of the GEMINI database. The second GEMINI query step has the same settings as the above step except for: “Additional constraints expressed in SQL syntax”: somatic_status = 2 AND somatic_p_value <= 0.05 AND filter IS NULL where we told GEMINI to to include only variants that have a p_value of less than 0.05. The third GEMINI query run had the same settings as step two, except that ‘Additional columns (comma-separated)’ had type, gt_alt_freqs.TUMOR, gt_alt_freqs.NORMAL, ifnull(nullif(round(max_aaf_all,2),-1.0),0) AS MAF, gene, impact_so, aa_change, ifnull(round(cadd_scaled,2),'.') AS cadd_scaled, round(gerp_bp_score,2) AS gerp_bp, ifnull(round(gerp_element_pval,2),'.') AS gerp_element_pval, ifnull(round(hs_qvalue,2), '.') AS hs_qvalue, in_omim, ifnull(clinvar_sig,'.') AS clinvar_sig, ifnull(clinvar_disease_name,'.') AS clinvar_disease_name, ifnull(rs_ids,'.') AS dbsnp_ids, rs_ss, ifnull(cosmic_ids,'.') AS cosmic_ids, ifnull(overlapping_civic_url,'.') AS overlapping_civic_url, in_cgidb

After we had generated a report for somatic variants with some filters, we also tried to generate a report of genes affected by the variants based on the somatic variant report that we generated already. Such a report would include annotations that apply to the gene affected by the variant and not the variant itself. While some of this information is built in the GEMINI database, they are stored in a different table called ‘gene-detailed’ separated from the variant table that we have been exploring. Fortunately, we can join the the two tables to access information from them in one query, but the basic query constructor option won’t do this, hence we ran GEMINI query again but this time, the ‘Build GEMINI query using’ option was set to Advanced query constructor. “The query to be issued to the database” was set to SELECT v.gene, v.chrom, g.synonym, g.hgnc_id, g.entrez_id, g.rvis_pct, v.clinvar_gene_phenotype FROM variants v, gene_detailed g WHERE v.chrom = g.chrom AND v.gene = g.gene AND v.somatic_status = 2 AND v.somatic_p_value<= 0.05 AND v.filter IS NULL GROUP BY g.gene and the “Genotype filter expression”: gt_alt_freqs.NORMAL <= 0.05 AND gt_alt_freqs.TUMOR >= 0.10 was still used.

The aim of including extra annotations to the GEMINI-generated gene report (that is, the output of the last GEMINI query) is to make interpreting the final output easier. Unfortunately, while GEMINI-annotate allows us to add specific columns to the variant table of the database we created, it does not allow us to include additional annotations into the gene_detailed table. Hence we added our extra annotations to our generated gene-centered report. We did this by joining the gene-centered report to the tabular annotation sources that we imported (Uniprot_Cancer_Genes.txt file (column 1), CGI.txt file (column 1) and CIViC Genesummarie.tsv file(column 3)) using the Join two files tool on Galaxy. For each join run, we used the output of the previous step, that is, the output of the last GEMINI query run was the input for Uniprot and the output of the uniprot join run was the input for CGI etc. The value to put for empty fields for uniprot and CGI join run was set as ‘0’ and ‘.’ for CIViC. The Uniprot data used 1 and 0 to indicate if the gene is a proto-oncogene or not or a tumor suppressor gene or not. We used this information to say that for the genes missing from the Uniprot annotation dataset, we want to fill the corresponding two columns of the join result with 0 to indicate the common case that a gene affected by a variant is neither a known proto-oncogene (is_OG), nor a tumor suppressor (is_TS) gene. For CGI biomarkers, we also used 0 and 1 to indicate if the gene affected by the variant is a cancer biomarker or not. For the CIViC join, the run was supposed to add the gene id column to our report and indicate the gene number and where it is not found, put a full stop (.).

The final operation that was carried out on our report was the rearrangement operation using the Column arrange by header name and the last output of the Join operation was selected in the “file to arrange”, that is the input. The columns to be specified by name are: gene, chrom, synonym, hgnc_id, entrez_id, rvis_pct, is_OG, is_TS, in_cgi_biomarkers, clinvar_gene_phenotype, gene_civic_url, and description in this order and we discarded unspecified columns.

Column Abbreviation	Full Meaning
gene	The gene name
chrom	The chromosome on which the gene resides
synonym	Other gene names (previous or synonyms) for the gene
hgnc_id	The HGNC identifier for the gene if HGNC symbol is TRUE
entrez_id	The entrez gene identifier for the gene
rvis_pct	The RVIS percentile values for the gene
is_OG	Is a proto-oncogene
is_TS	Is a tumor suppressor gene
in_cgi_biomarkers	Is a cancer biomarker
clinvar_gene_phenotype	delimited list of phenotypes associated with this gene (includes any variant in the same gene in clinvar not just the current variant)
gene_civic_url	The url to CIViC
description	The description of the mutation

Conclusion

This project showed the importance of computational tools in detecting and differentiating between types of variants or mutations and it makes it an applicable tool in processes that require high precision such as detecting unannotated mutations. The result of this project provided 41 genes with somatic variations and are not likely to be false positives. Detailed results showed that genes APC on chromosome 5 and TP53 on chromosome 17 are tumor suppressor genes and linked to CGI biomarkers. While genes ELAC2 and RNF213 on chromosome 17 and KRAS on chromosome 12 are proto oncogenes. There were two variations described to be associated with malignant tumors on chromosome 5. Studies has shown that mutations in the VCAN gene have been associated with Wagner syndrome, a condition that leads to progressive vision loss starting in childhood or early adulthood. TP53 is a tumor suppressor gene but when mutation occurs in it such as seen here, it may lead to the development of cancers and production of cancer biomarkers.

Here is the link to the github file containing the tabular report:
https://github.com/Aanuoluwaduro/HackBioStage-Two-Task/blob/main/Final%20Tabular%20Report.tabular
Here is the link to the repository itself containing all the detailed results.
https://github.com/Aanuoluwaduro/HackBioStage-Two-Task

Here is the link to the task that was reproduced:
https://github.com/Fredrick-Kakembo/Somatic-and-Germline-variant-Identification-from-Tumor-and-normal-Sample-Pairs/blob/main/README.md#variant-annotation-and-reporting