

vcfView

Brian O'Sullivan
15th June 2020

In This Vignette

- About this vignette
- Download and Installation
- Data
- Single VCF exploratory analysis with vcfView
- Dataset overview with vcfView
- Adding extensible components to vcfView
- Usage Notes relating to The Texas Cancer Research Biobank Dataset

About this vignette

This vignette uses vcfView to analyse the mutational processes and signatures at work within a region of interest in the somatic allele frequency spectrum for an individual VCF file. It breaks down candidates that have passed or failed individual Mutect2 filters and at each step shows how to analyse their impact on protein. It also describes how to generate a summary of all vcf files within a dataset for analysis to identify patterns which may exist across the set.

Download and Installation

There are two options to install vcfView in Rstudio.

i) Install vcfView and open with Rshiny.

If you have already installed shiny you can use setwd() to go to the required install destination directory and from the Rstudio console run,

```
> shiny::runGitHub('vcfView', 'BrianOSullivanGit', destdir = getwd())
```

ii) Alternatively you can download the source from GitHub in a command tool with,

```
$ git clone https://github.com/BrianOSullivanGit/vcfView
```

Then in Rstudio open and run 'app.R' located in the vcfView directory.

Data.

This vignette uses VCF files generated from The Texas Cancer Research Biobank (TCRB) data. See section "VCF generation and usage notes" for full details and conditions.

Single VCF exploratory analysis.

We will begin by analysing the VCF of a single donor from the dataset, TCRBOA7. Open vcfView and click on select VCF, navigate to the Data subdirectory and open TCRBOA7.filtered.vcf.gz.

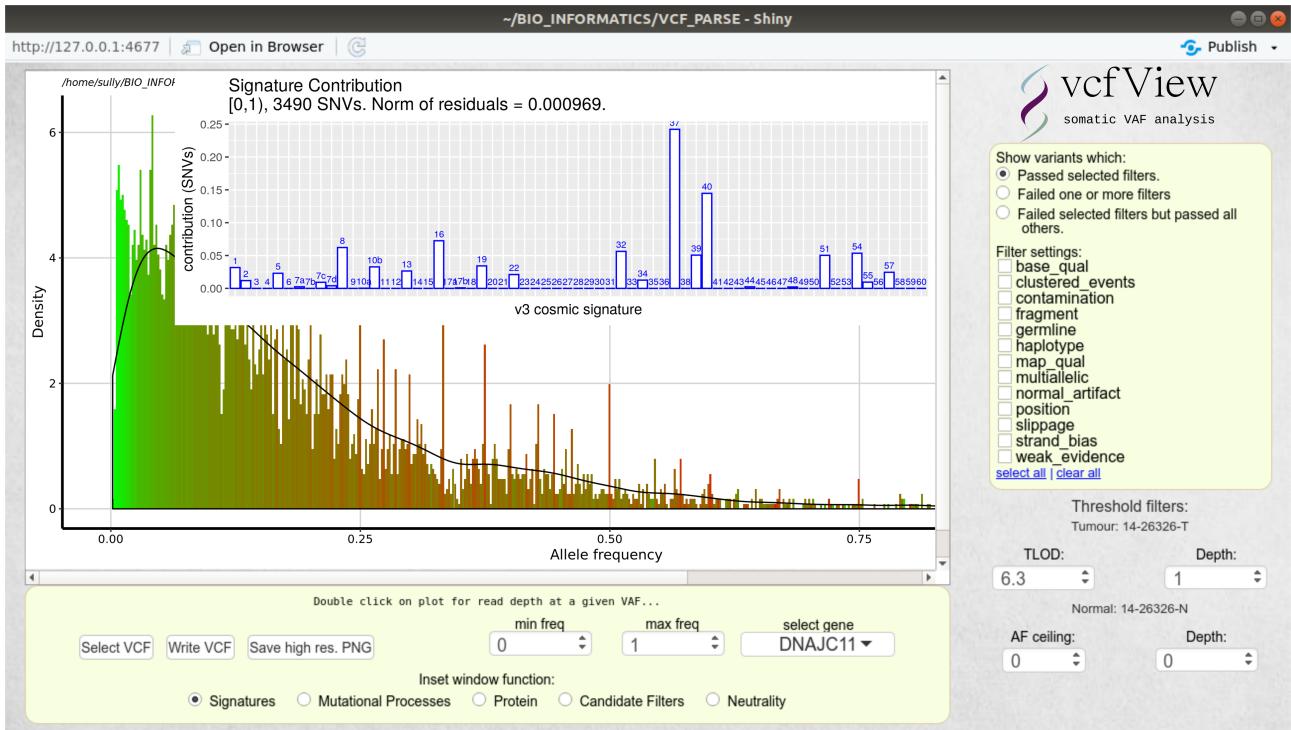


Figure 1: vcfView displaying signatures inset function.

When loaded a VAF density plot is displayed in the main window (Figure 1). Bins are coloured using a custom pallet ranging from green to red, providing logarithmic colour scaled interpolation with median sequencing depth in the tumour for each bin. The actual tumour depth at a given bin may be displayed under the density plot by double clicking on the required bin. Next click on 'select all' under 'Filter settings' on the panel to the right. This density plot now displays all variants annotated as 'PASS' in the VCF (ie. which passed all Mutect2 filters).

Inset window functions are listed at the bottom of the screen. The selected function is triggered whenever a frequency range from variant allele frequency spectrum is set (either by manually entering min and max freq values in the UI or by 'click and drag' selection directly from the plot). Select the 'Signatures' function and set the min and max freq. values to '0' and '1' respectively. A bar plot representing the optimal contribution fit from each of the COSMIC v3 reference signatures required to reconstruct the 96 mutational class profile contained within this VCF is displayed in the inset plot.. Alternatively, the 96 tri-nucleotide context plot for the SNVs within this allele frequency range may be displayed by selecting 'Mutational Processes' from the inset window function.

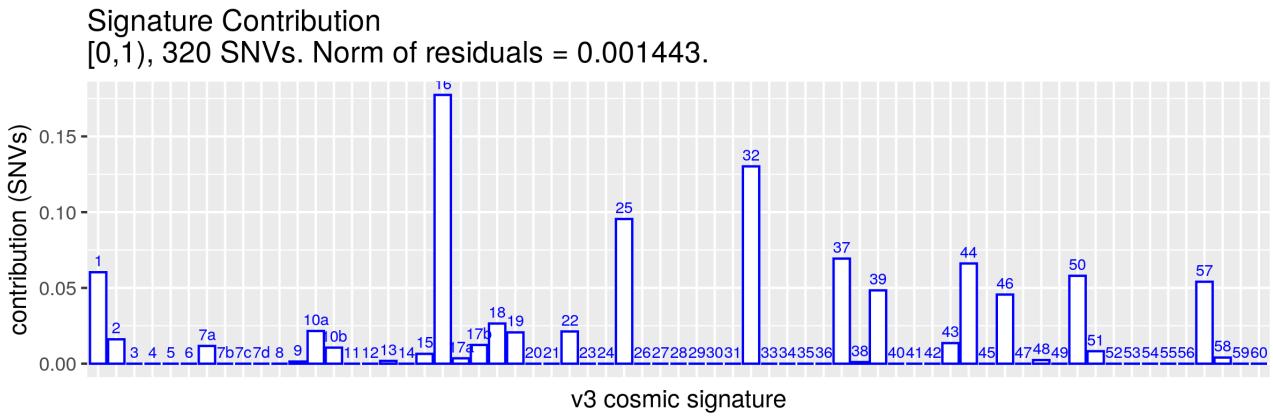


Figure 2: Cosmic v3 signatures fit for TCRBOA7.

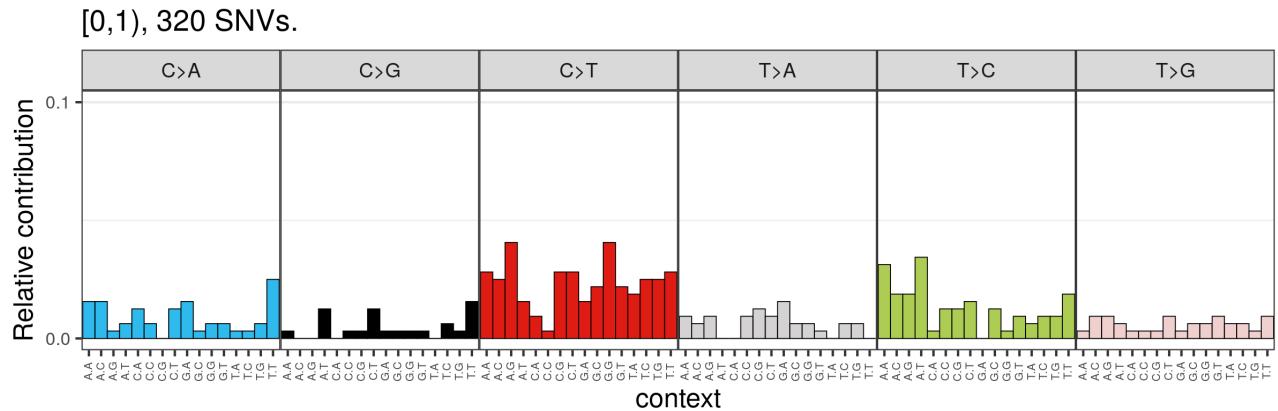


Figure 3: TCRBOA7 mutational processes breakdown against with the v3 signatures were fitted.

This sample, resected from a treatment naive, follicular lymphoma patient demonstrates a strong signal for COSMIC signature 25, a signature present in Hodgkin lymphoma cell lines (<https://cancer.sanger.ac.uk/cosmic/signatures/SBS/SBS25.tt>). Also present are signatures 16, 37 (aetiology unknown), 32 (Prior treatment with azathioprine to induce immunosuppression), 44 (defective DNA repair) and others relating to possible sequencing artefacts. Double clicking on or above the bar on any of the signatures displayed will open a link describing the signature on the Sanger Institute website. The sample is also noted as having high tumour cellularity (~90%). Without accounting from CNV this would perhaps indicate clonal variants being centered closer to 0.5 on the allele spectrum. Using the mouse and with the 'Signatures' function active, select a region (click & drag) of approx 0.1 in width centered around 0.2 on the spectrum. Next click, hold and drag move this window to the right along the spectrum and observe how the signature contributions change. You will notice a peak for signature 25 around the frequency window 0.34 to 0.44.

In this region select the 'Protein' function under 'Inset Window function'. When the frequency range is set and the 'Protein' function selected the 'select gene' list displays only genes impacted by coding variants within the subset selected. Again we are focusing on the frequency range 0.34 to

0.44. Click on 'select gene' to examine the protein impacts to each gene listed. The dropdown gene list shows there are four genes impacted by coding variants within this region of the allele spectrum annotated as PASS by Mutect2. One at a time, select each gene from the list. The inset window 'protein' function shows two carry synonymous variants. The others are a nonsense mutation near the start of KMT2D and a nonsynonymous variant (V3403M) in HSPG2.

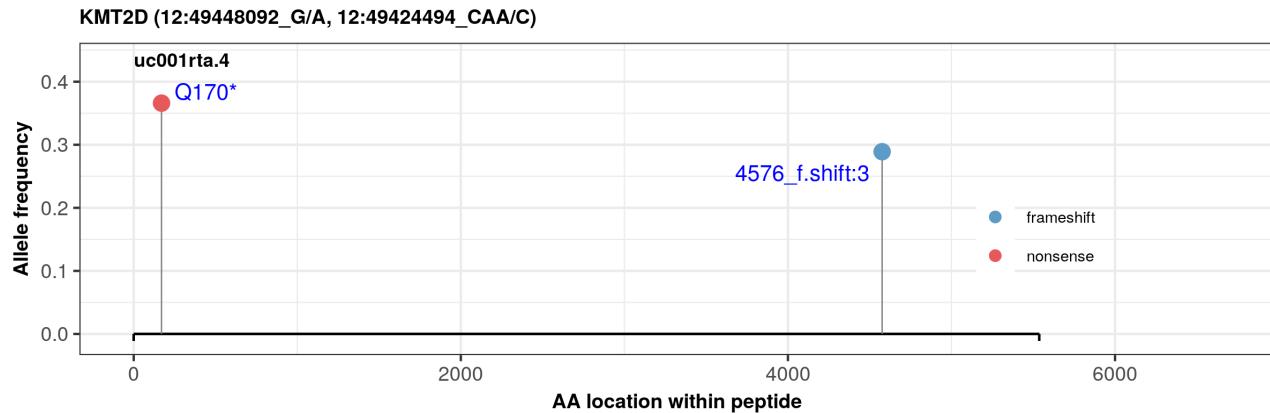


Figure 4: KMT2D mutations in TCRBOA7

KMT2D (mutated in this sample and shown in Figure 4 above) is the most recurrently mutated chromatin modifying gene in FL (~72% of cases) and a hallmark of the disease (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5805487/>). The majority of these mutations are nonsense (as in TCRBOA7) or frameshift events that result in a loss of the KMT2D protein. . To clear the frequency range (reset to min 0, max 0) double click on the density plot (outside the inset plot). Double clicking again will close the inset plot. Leave the 'select gene' set to KMT2D and manually set the min and max freq. to 0 and 1. This highlights another frameshift mutation in KMT2D for this donor just below 0.3 on the spectrum. Look at other parts of the spectrum for other relevant variants. For example, continue to slide the AF window selected to the right along the x-axis and look for other regions enriched for signature 25. Alternatively set your frequency window from 0.2 to 0.7 and examine the 'protein' inset function for relevant variants. You will notice other variants that have been linked to FL (for example NFE2L3, CREBBP).

Return to the 'signatures' inset plot. Set the frequency range back to 0.34 to 0.44 and again select the 'signatures' inset function. Click 'failed one or more filters' on the filter panel to the right to get an idea what the background signal looks like within this frequency range. The signature contribution plot shows a strong background component for signature 32. Click 'Passed selected filters' again and one at a time, deselect each Mutect2 filter. You will notice a significant jump in Signature 32 for the 'map_qual' filter. Read the tooltip associated with this filter and look up its meaning in Mutect2 documentation. Clear all filters and select only the 'map_qual' filter. Next select the option to display variants which 'Failed selected filters but passed all others'. The signature contribution plot displays a strong background component for signature 32. Select 'Passed selected filters' and 'select all' in the filter settings. Change the MMQ (median mapping quality) in the threshold filters (bottom right) to a more stringent cut-off of 40. You will notice a 35% decrease in signal 32 and a 40% decrease in signal 16.

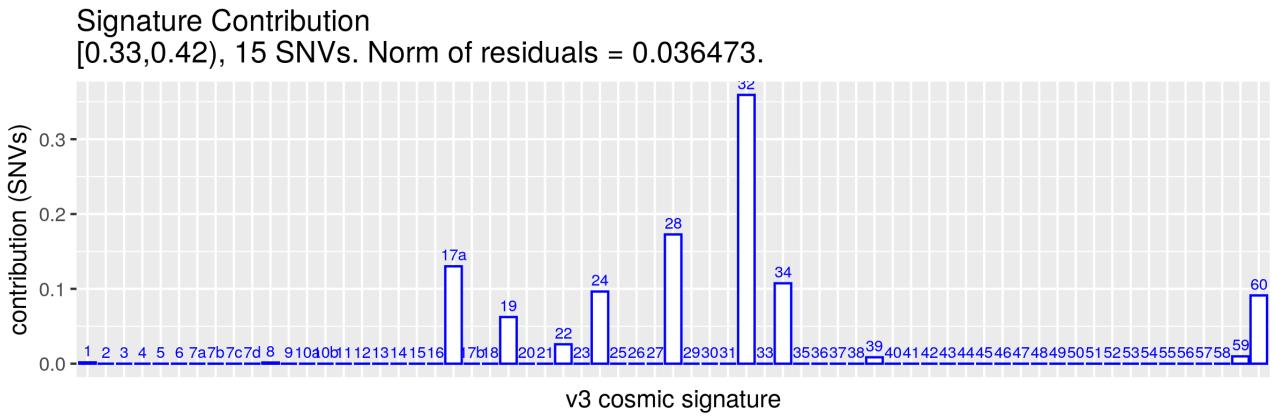


Figure 5: Signatures fit for variants which failed 'map_qual' filter within specified frequency range.

Finally, have a look at the filter breakdown for this VCF with the 'candidate filters' inset function. Set the frequency range to 0,1 and select 'candidate filters' and 'clear all'. The plot displays the fraction of variants that failed each filter. The majority of excluded variants failed the 'normal_artifact' filter. Clicking 'normal_artifact' in the filter panel will remove this filter from the plot allowing us to focus on the remaining filters. To see, for example, which filters are excluding candidate variants at low allelic frequency select a narrow frequency range at the start of the spectrum. This shows the majority of excluded variants within this allele frequency range failed the 'clustered_events' filter.

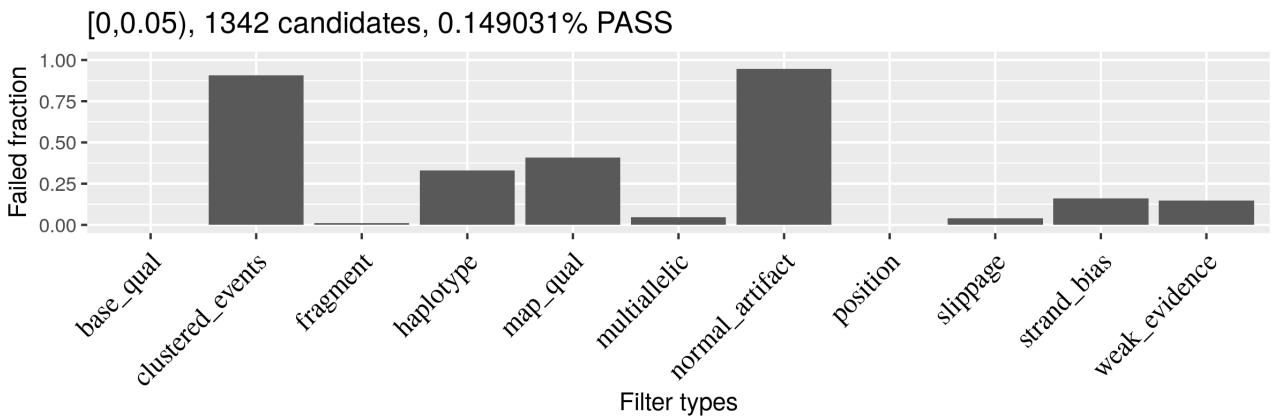


Figure 6: Breakdown of which filters low frequency candidate variants are failing.

Dataset overview with vcfView.

Creating a summary file.

A summary file is a concatenation of variant records across a number of VCF files. For example, all variants within the dataset from the same chromosome, or location within a chromosome. Alternatively it may contain all variants that failed only a particular filter or group of filters or that passed all filters.

Currently, a summary file is created on the command line. We hope to provide this functionality directly from within vcfView in a future release.

To create a summary file first copy metadata from one of the VCFs in the set.

```
$ grep '^(##|##CHROM)' example.vcf > dataset.summary
```

For example, to create a summary containing all records which were annotated as 'PASS',

```
$ grep -P '\tPASS\t' *.vcf >> dataset.summary
```

Lets apply this to our dataset. Cd to the 'Data' directory containing the VCF files.

```
$ zgrep '^(##|##CHROM)' TCRBOA7.filtered.vcf.gz > dataset.summary
```

(note: if your VCF files are not compressed, use grep/egrep instead of zgrep/zgrep)

Next extract and append to it the records of interest from the dataset concerned. Here, we will determine if we can identify any potential TiN (tumour in normal) variants in our dataset. For this we want to extract all variant records that have failed normal_artifact, panel_of_normals or germline filters. We will also include variants which passed all filters in our dataset. To achieve this we execute the following commands.

```
# Add variants which passed all filters.  
zcat TCRBOA[1-7].filtered.vcf.gz | egrep -v '^(#)' | grep -P '\tPASS\t'  
>> dataset.summary  
  
# Add germline filtered variants.  
zcat TCRBOA[1-7].filtered.vcf.gz | egrep -v '^(#)' | grep -P '\tgermline\t' >> dataset.summary  
  
# Add normal_artifact filtered variants.  
zcat TCRBOA[1-7].filtered.vcf.gz | egrep -v '^(#)' | grep -P '\tnormal_artifact\t' >> dataset.summary  
  
# Add a combination of both.  
zcat TCRBOA[1-7].filtered.vcf.gz | egrep -v '^(#)' | grep -P '\tgermline;normal_artifact\t' >> dataset.summary
```

The combination of zcat and egrep -v '^(#)' above extracts all the variant records from the dataset (i.e. all lines that do not start with a '#' metadata marker). The result is piped to the next grep command which filters out only variant records annotated with the filter of interest and appends them to our summary.

Close vcfView and open it again this time selecting the 'dataset.summary' file you have just created. Select the 'normal_artifact' filter and protein inset function. Set the depth threshold filter (below the panel on the right) for both tumour and normal samples to 20. Set the AF ceiling for normal to 0.05. This will filter out all variants for which the allele frequency in normal is above 0.05.

Next, select an allele frequency window from about 0.2 to 1. We are searching for variants flagged as allele in normal for which the allele frequency in the tumour is significantly greater than in the normal, and for which there is good coverage at that locus in both tumour and normal. No variants are found in the protein inset function window. Slowly increase the AF ceiling (click the 'up' arrow) until some protein impacting variants come into view. A variant in 'ACAN' with allele frequency in the normal of ~0.075 and ~0.28 in the tumour is the first variant found. To find the VCF file containing this variant record note the locus from the protein plot title and use 'grep' to locate the file in question. Loci (or rsid if the VCF contains such annotation) are listed in the order in which they appear on the plot (from left to right). No protein annotated variants failing only normal filters were found with a large disparity between tumour and normal allele frequency. VcfView did not highlight any putative tumour in normal variants in this dataset.

Adding extensible components to vcfView

VcfView is extensible allowing other algorithms to take advantage of its VCF preprocessing capabilities. This section describes how to integrate software into vcfViews inset function to take advantage of these capabilities. Software is integrated into vcfView by adding to the exten.R file which is sourced by vcfView on load. The user adds three things to this file to perform integration. First lines are added to load any external libraries required by their software. Next define a string variable of the form,

```
exten_<function id>="<a brief string which will appear under the inset window function>"
```

Finally add a hook function which will be called by vcfView when the new inset function is triggered. This function takes 3 arguments.Two relating to allele frequency range. The third is a data.frame containing VCF data presubsetted by vcfView according to filter and threshold settings at the time the function was triggered.

To illustrate this process we have integrated the neutralitytestr library as an inset into vcfView . The code to do this is listed and described below and its purpose is solely to demonstrate the integration process. The relevant extract from extern.R follows. Appending these lines to the extrn.R file will add this processing to vcfView.

```
#  
# Extensible hook for "neutralitytestr"  
  
if (!requireNamespace("neutralitytestr", quietly = TRUE))  
  BiocManager::install("neutralitytestr")  
  
library(neutralitytestr)  
  
exten_str_Nevol="Neutrality"  
  
exten_fn_Nevol <- function(result, lower_frange, upper_frange)  
{  
  # Subset out all the SNVs within required allele freq. range.  
  snvIdx = which(result$AF > lower_frange &  
                 result$AF <= upper_frange)  
  
  n <- neutralitytest(result$AF[snvIdx], fmin = lower_frange, fmax = upper_frange)  
  
  return(lsq_plot(n))  
}
```


VCF generation and usage notes

VCF Generation

In this vignette we demonstrate vcfView using publicly available data from The Texas Cancer Research Biobank (TCRB). Full details and access are described in the associated paper (<https://www.nature.com/articles/sdata201610>). This dataset was chosen as it is open access data from tumor and normal matched pairs. We thank the donors and authors for making this data publicly available for the advancement of research. Usage notes associated with the dataset are at the end of this section.

From this dataset we generated 7 matched tumour, normal somatic VCF files using Mutect2 GATK4 (4.1.7.0-25, built from head of tree at time of writing) which we have included in the 'Data' directory of this vignette. An overview of the process we used to create one of these sample pairs (TCRBOA7) is shown below for reference. The reader should refer to the latest version of the GATK best practices should they want to rebuild these VCFs.

```
# BAM files were downloaded according to instructions provided in relevant article
# (https://www.nature.com/articles/sdata201610)
#
# Mutect2 and samtools installed previously.
#
# The user may wish to run complementary Mutect2 filtering for sequence
# context-dependent artifacts
# (e.g. OxoG or FFPE deamination,
# https://javadoc.io/static/org.broadinstitute/gatk/4.1.4.1/org/broadinstitute/
# hellbender/tools/exome/FilterByOrientationBias.html).
# It has not been run here.
# No panel of normals was used when creating these VCFs.
# If rebuilding with a panel of normals remember to account for
# the appropriate filter when building a summary file.

DONOR_NUM=${1}
TEXAS_DATASET_PREFIX="TCRBOA"
TUMOUR_BAM_NAME=${TEXAS_DATASET_PREFIX}${DONOR_NUM}_T_WEX.bam
NORMAL_BAM_NAME=${TEXAS_DATASET_PREFIX}${DONOR_NUM}_N_WEX.bam

BAM_LOCATION="<..insert your BAM location here..>/Texas/Bams"
VCFS_LOCATION="<..insert your VCF location here..>/Texas/Vcf"
JAR_LOCATION="<..insert your jar location here..>/gatk-package-4.1.7.0-25-g3dabc3f-
SNAPSHOT-local.jar"
REF_LOCATION="<..insert your reference location here..>/Texas/Ref/GRCh37-lite.fa"

TUMOUR_SAMPLE_NAME=`samtools view -H ${BAM_LOCATION}/${TUMOUR_BAM_NAME} | grep
'^@RG' | sed "s/.SM:[\^t]*\).*/\1/g" | uniq`'
NORMAL_SAMPLE_NAME=`samtools view -H ${BAM_LOCATION}/${NORMAL_BAM_NAME} | grep
'^@RG' | sed "s/.SM:[\^t]*\).*/\1/g" | uniq`'

# Run Mutect2
java -Xmx4g -Djava.io.tmpdir=/data4/bosullivan/Texas/MutectTmpDir \
-jar ${JAR_LOCATION} Mutect2 \
-R ${REF_LOCATION} \
-I ${BAM_LOCATION}/${TUMOUR_BAM_NAME} \
-I ${BAM_LOCATION}/${NORMAL_BAM_NAME} \
-tumor ${TUMOUR_SAMPLE_NAME} \
-normal ${NORMAL_SAMPLE_NAME} \
-tumor-lod-to-emit 0 \
-O ${VCFS_LOCATION}/${TEXAS_DATASET_PREFIX}${DONOR_NUM}.vcf.gz \
-bamout ${VCFS_LOCATION}/${TEXAS_DATASET_PREFIX}${DONOR_NUM}_bamout.bam 2>&1

# Filter calls.
java -Xmx4g -jar ${JAR_LOCATION} FilterMutectCalls \
-R ${REF_LOCATION} \
-V ${VCFS_LOCATION}/${TEXAS_DATASET_PREFIX}${DONOR_NUM}.vcf.gz \
-O ${VCFS_LOCATION}/${TEXAS_DATASET_PREFIX}${DONOR_NUM}.filtered.vcf.gz
```

These VCFs are located in the Data subdirectory of vcfView and are used to demonstrate tool utility in this vignette.

Usage Notes relating to The Texas Cancer Research Biobank Dataset

This usage note is taken from the paper relating to the Texas Cancer Research Biobank Dataset (<https://www.nature.com/articles/sdata201610>).

By downloading or utilizing any part of this dataset, end users must agree to the following conditions of use:

- No attempt to identify any specific individual represented by these data or any derivatives of these data will be made.
- No attempt will be made to compare and/or link this public data set or derivatives in part or in whole to private health information.
- These data in part or in whole may be freely downloaded, used in analyses and repackaged in databases.
- Redistribution of any part of these data or any material derived from the data will include a copy of this notice.
- The data are intended for use as learning and/or research tools only.
- This data set is not intended for direct profit of anyone who receives it and may not be resold.
- Users are free to use the data in scientific publications if the providers of the data (Texas Cancer Research Biobank and Baylor College of Medicine Human Genome Sequencing Center) are properly acknowledged.

Implementation of common vocabularies facilitates semantic interoperability and data reuse. Annotations for OA cases include controlled terminologies for race, gender and ethnicity from the NIH; pathological diagnosis from World Health Organisation's ICD-O-3 (International Classification of Diseases-Oncology version 3); and tumor stage and grade data from Union for International Cancer Control (UICC/AJCC). Use of standard metadata facilitates syntactic interoperability. The TCRB used standard data elements, file formats and metadata. Because OA data can be downloaded and re-shared on the provision that the conditions of use are included and abided by, all annotations were also included as comments in BAM file headers with a reference to the conditions of use. These metadata were included to ensure that the clinical and pathological data cannot become decoupled from the sequence data.