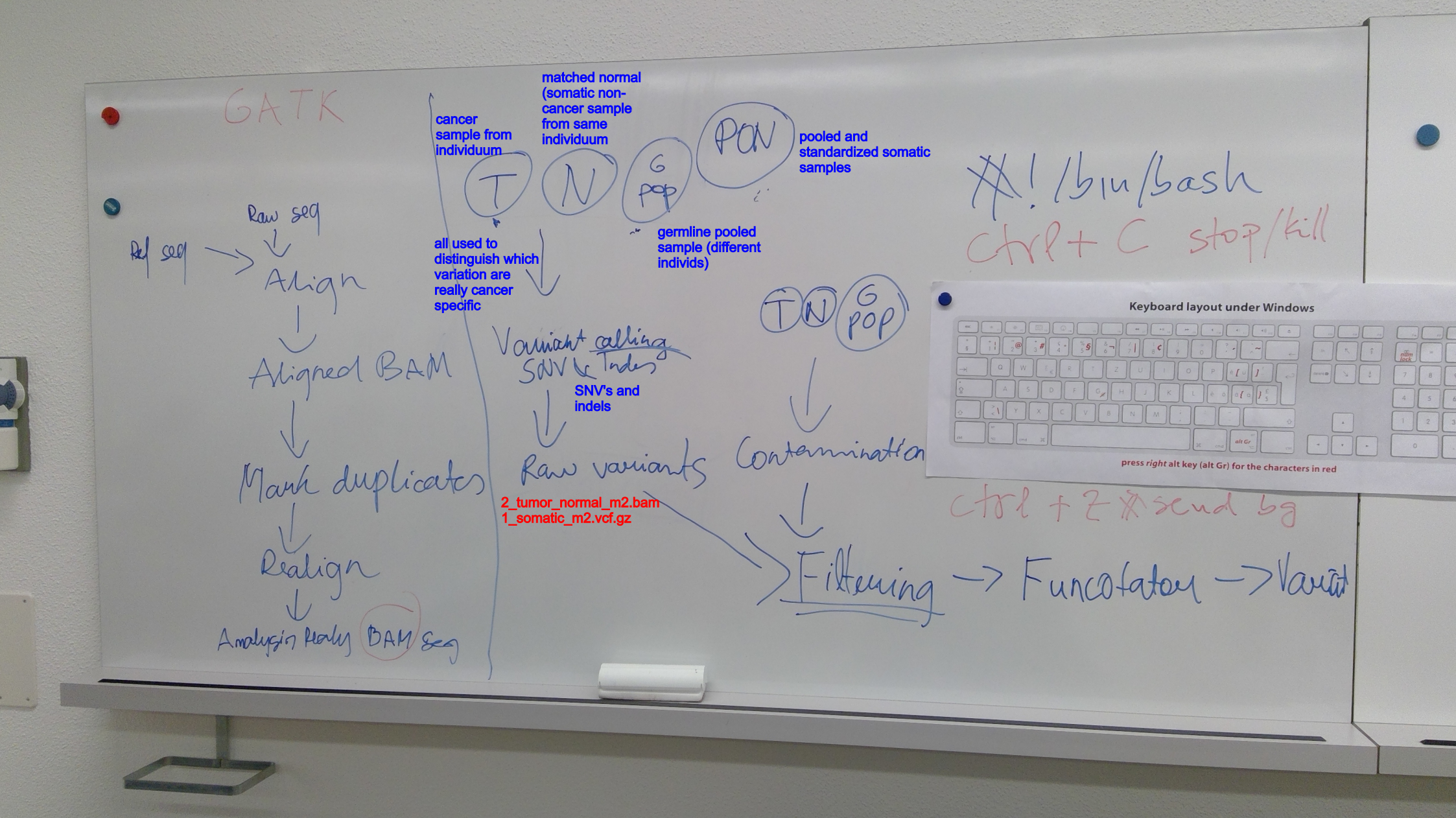

GATK
cancer sample from individuum
matched normal (somatic non-cancer sample from same individuum
pooled and standardized somatic samples
germline pooled sample (different individs)
all used to distinguish which variation are really cancer specific
Raw seq
Ref seq
Align
Aligned BAM
Mark duplicates
Realign
Analysis Ready BAM Seq
Variant calling
SNV & Indels
SNV's and indels
Raw variants
2_tumor_normal_m2.bam
1_somatic_m2.vcf.gz
Contamination
Filtering
-> Funcofator
-> Varia
#!/bin/bash
ctrl+C stop/kill
ctrl+Z send bg
Keyboard layout under Windows
press right alt key (alt Gr) for the characters in red

2.1 Call somatic SNVs and indels and generate a BAMOUT

We start by calling somatic short mutations on our HCC1143 tumor sample and matched normal using Mutect2. This command produces:

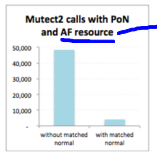
1. A raw unfiltered somatic callset restricted to the specified intervals list
2. A BAM containing reassembled alignments
3. Mutect stats file named `/home/jupyter/notebooks/sandbox/1_somatic_m2.vcf.gz.stats`

```
!gatk Mutect2 \
    -R /home/jupyter/notebooks/ref/Homo_sapiens_assembly38.fasta \
    -I /home/jupyter/notebooks/bams/tumor.bam \
    -I /home/jupyter/notebooks/bams/normal.bam \
    -normal HCC1143_normal \
    -pon /home/jupyter/notebooks/resources/chr17_m2pon.vcf.gz \
    --germline-resource /home/jupyter/notebooks/resources/chr17_af-only-gnomad_grch38.vcf.gz \
    -L /home/jupyter/notebooks/resources/chr17plus.interval_list \
    -O /home/jupyter/notebooks/sandbox/1_somatic_m2.vcf.gz \
    -bamout /home/jupyter/notebooks/sandbox/2_tumor_normal_m2.bam
```

reference
matched normal
PON
g pop (only chr17)
output
interval

➤ **What is the value of using a matched normal control?**

without having a non-tumour cell sample from the same indiv. we would call way to much variation as cancer var instead of individ. specific var



= g pop

Mutect2 uses the matched normal to additionally **exclude rare germline variation** not captured by the germline resource and individual-specific artifacts.

Mutect2 uses a germline population resource towards evidence of alleles being germline. The simplified sites-only gnomAD resource retaining allele-specific frequencies is available at ftp://gsapubftp-anonymous@ftp.broadinstitute.org/bundle/Mutect2.

A panel of normals (PoN) has a vital role that fills a gap between the matched normal and the population resource. Mutect2 uses the PoN to catch additional **sites of noise in sequencing data**, like mapping artifacts or other somewhat random but systematic artifacts of sequencing and data processing. **technically similar samples that were sequenced on the same platform.**

3.1.1 Run `GetPileupSummaries` to summarize read support for a set number of known variant sites.

The tool tabulates read counts that support REF, ALT and OTHER alleles for the sites in the resource. This involves a known germline variant resource to **limit analysis to sites that are commonly variant**. Use a population germline resource containing only common biallelic variants

Here we use a human population germline resource, gnomAD. Let's run the tool on the tumor and the normal.

```
# Run on Tumor samples
!gatk GetPileupSummaries \
    -I /home/jupyter/notebooks/bams/tumor.bam \
    -V /home/jupyter/notebooks/resources/chr17_small_exac_common_3_grch38.vcf.gz \
    -L /home/jupyter/notebooks/resources/chr17_small_exac_common_3_grch38.vcf.gz \
    -O /home/jupyter/notebooks/sandbox/7_tumor_getpileupsummaries.table
```

```
!head /home/jupyter/notebooks/sandbox/7_tumor_getpileupsummaries.table
```

```
#<METADATA>SAMPLE=HCC1143_tumor
contig  position        ref_count       alt_count       other_alt_count allele_frequency
chr6    29942512        9       0       0       0.063
chr6    29942517        13      1       0       0.062
chr6    29942525        13      7       0       0.063
chr6    29942547        36      0       0       0.077
chr6    29942563        19      23      0       0.086
chr6    29942594        36      7       0       0.118
chr6    29942625        31      0       0       0.125
chr6    29942642        22      3       0       0.114
```

alt_count
other_alt_count
allele_frequency

```
In [13]:  !head /home/jupyter/notebooks/sandbox/7_normal_getpileupsummaries.table
```

```
#<METADATA>SAMPLE=HCC1143_normal
contig  position        ref_count       alt_count       other_alt_count allele_frequency
chr6    29942512        7       4       0       0.063
chr6    29942517        12      4       0       0.062
chr6    29942525        13      7       0       0.063
chr6    29942547        27      0       0       0.077
chr6    29942563        19      14      0       0.086
chr6    29942594        21      6       0       0.118
chr6    29942625        22      0       1       0.125
chr6    29942642        11      2       0       0.114
```

Each command produces a six-column table as shown. The `alt_count` is the count of reads that support the ALT allele in the germline resource. The `allele_frequency` corresponds to that given in the germline resource. Counts for `other_alt_count` refer to reads that support all other alleles.

The tool considers *homozygous-variant* sites in the sample where the alternate allele frequency (AF) in the population resource ranges between 0.01 and 0.2. This range is adjustable. We can expect a lot of contamination by alternate alleles at sites where the alternate AF is large, so those sites wouldn't tell us much. Conversely, at homozygous-alternate sites where the variant allele is rare in the population, we are more likely to observe the presence of REF or other alleles if there was cross-sample contamination, and therefore we will be able to measure contamination more accurately.

shows the aligned reads of both the tumour and the norm

3.1.2 Estimate contamination with `CalculateContamination`.

The tool gives the fraction contamination. This estimation informs downstream filtering by FilterMutectCalls.

```
!gatk CalculateContamination \
    -I /home/jupyter/notebooks/sandbox/7_tumor_getpileupsummaries.table \
    -tumor-segmentation /home/jupyter/notebooks/sandbox/segments.table \
    -O /home/jupyter/notebooks/sandbox/8_tumor_calculatecontamination.table
```

```
!cat /home/jupyter/notebooks/sandbox/8_tumor_calculatecontamination.table
```

```
sample  contamination   error
HCC1143_tumor   0.019127415053507495    0.0021541303669512504
```

```
!cat /home/jupyter/notebooks/sandbox/8_pair_calculatecontamination.table
```

```
sample  contamination   error
HCC1143_tumor   0.011485364960150258    0.0019180421331441303
```

For our small tumor BAM file, you can see the contamination is ~0.0191 with an error of ~0.0022. We get a slightly lower number, ~0.0120 +/- 0.00454 for the matched estimate. For the full BAM file, we see a slightly larger contamination number. This threshold informs you to be wary of calls with less than that number for the alternate allele fraction.

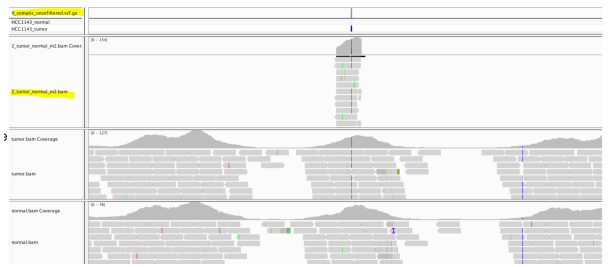
3.2 Apply filters with `FilterMutectCalls`

In this step, we filter the small data, 1_somatic_m2.vcf, with `FilterMutectCalls`. The tool **uses the annotations within the callset, and if provided, uses the contamination table** in filtering. Default settings are tuned for human somatic analyses.

```
!gatk FilterMutectCalls \
    -R /home/jupyter/notebooks/ref/Homo_sapiens_assembly38.fasta \
    -V /home/jupyter/notebooks/sandbox/1_somatic_m2.vcf.gz \
    --contamination-table /home/jupyter/notebooks/sandbox/8_pair_calculatecontamination.table \
    --stats /home/jupyter/notebooks/sandbox/1_somatic_m2.vcf.gz.stats \
    --tumor-segmentation /home/jupyter/notebooks/sandbox/segments.table \
    -O /home/jupyter/notebooks/sandbox/9_somatic_oncefiltered.vcf.gz
```

reference
raw variants
contamination
stats file
output
used in contamination calc as well

This produces a VCF callset `9_somatic_oncefiltered.vcf.gz` and index. Calls that are **likely true positives get the PASS label** in the FILTER field, and calls that are likely false positives are labeled with the reason(s) for filtering in the FILTER field of the VCF. We can view the available filters in the VCF header using



➤ **We see a C→T variant light up in red for the tumor but not the normal. What do you think is happening in 2_tumor_normal_m2.bam?** we observe that there is good reason that this specific call passed the filtering process and shows in the once_filtered.vcf.gz

➤ **What does the coverage tell you?**

very deep coverage and hence good trust in call

➤ **What are the three grouped tracks for the bamout? What do the colors indicate? What differentiates the pastel versus gray reads?**

colours represent different haplotypes

➤ **How do you feel about this somatic call?**

confident, clear call with good coverage

5 ANNOTATE MUTATIONS WITH FUNCOTATOR

Another approach to filtering mutation calls is by the significance of their functional impact. For example, a stop codon in the middle of a protein coding region or a missense mutation that changes how a protein functions is more significant than a silent mutation or a mutation in the middle of an intron.

To gauge functional impact, we must know which regions of the genome code for protein sequence and which correspond to elements important to gene expression. Transcript annotation resources such as GENCODE capture such information in a standardized General Transfer Format (GTF).

GATK4 Funcotator annotates variant alleles with information from any number of annotation resources. The annotation resources must be organized in a particular way. You can download prepared Funcotator resource bundles from `gs://broad-public-datasets/funcotator/` or use GATK4 `FuncotatorDataSourceDownloader` to download the latest data sources directly from your GATK4 install. For this tutorial, we have specially prepared a small annotation resource.

Annotate the `9_somatic_oncefiltered.vcf.gz` mutation callset with the resource.

```
!gatk Funcotator \
    --data-sources-path /home/jupyter/notebooks/resources/funcotator_dataSources_GATK_Workshop_20181205/ \
    --ref-version hg38 \
    -R /home/jupyter/notebooks/ref/Homo_sapiens_assembly38.fasta \
    -V /home/jupyter/notebooks/mutect2_precomputed/9_somatic_oncefiltered.vcf.gz \
    -O /home/jupyter/notebooks/sandbox/12_somatic_oncefiltered_funcotate.vcf.gz \
    --output-file-format VCF
```

ref
filtered variant calls

This produces a VCF callset with annotations. If needed, Funcotator can instead write results in historic Mutation Annotation Format (MAF) given `--output-file-format MAF`.

➤ **Examine the annotations for the TP53 mutation that we viewed earlier in IGV, at chr17:7674220.**

```
!zgrep chr17 /home/jupyter/notebooks/sandbox/12_somatic_oncefiltered_funcotate.vcf.gz | grep 7674220
```

We see an arginine (R) to glutamine (Q) missense mutation at position 248. In our 124 mutation records, 21 are annotated with MISSENSE, and of these, ten PASS filters.

```
# Count the total mutation records
!zgrep -v "^#" /home/jupyter/notebooks/sandbox/12_somatic_oncefiltered_funcotate.vcf.gz | wc
```

```
# Count the number of MISSENSE records
zgrep -v "^#" /home/jupyter/notebooks/sandbox/12_somatic_oncefiltered_funcotate.vcf.gz | grep "|MISSENSE|" | wc
```

```
# number of MISSENSE records that PASS filters
!" /home/jupyter/notebooks/sandbox/12_somatic_oncefiltered_funcotate.vcf.gz | grep "|MISSENSE|" | grep PASS | wc
```

Here are a few more filtered indel calls to explore.

CHROM	POS	REF	ALT	
chr17	26,982,033	G	GC	artifact_in_norma
chr17	35,671,734	CTT	C,CT,CTTT	artifact_in_nc
chr17	47,157,394	CAA	C,CAAA	artifact_in_norm
chr17	68,907,890	GA	G,GAA	artifact_in_normal;base_quality;germline_risk
chr17	69,182,632	C	CA	artifact_in_normal;c

all this indels didn't pass filters for reasons given here