DNA-Seq: Whole Genome Sequencing Variant Calling

Introduction

Variant calls from Whole Genome Sequencing (WGS) data are produced using pipelines distinct from those used for WXS and Targeted Sequencing samples. The GDC WGS variant calling workflows currently generate multiple downstream data types, including simple somatic mutations (SSMs), structural variants (SVs), and copy number variations (CNVs), leveraging the following software packages:

- Manta: Structural variants, which are available in both VCF and BEDPE format.
  - Additionally, Manta generates a set of candidate indels, which are subsequently used as input for the Strelka SSM calling pipeline to enhance Strelka's coverage across all indel sizes. Note that the Manta candidate indel file is not released on the GDC data portal.
- Strelka: Simple nucleotide variants, including both point mutations and Indels, which are available in VCF format.
- SvABA: Indel variants only, which are available in VCF format, and structural variants, which are available in both VCF and BEDPE format.
- GATK4 MuTect2: Simple nucleotide variants, including both point mutations and Indels, which are available in VCF format.
- GATK4 CNV: Copy number segments, which are available in TXT format.
  - Additionally, an auxiliary TAR file containing intermediate calling files is produced. This auxiliary file is intended for expert data reviewers to manually assess potential CNV model discrepancies and is not designed for regular user consumption.

Output from the new WGS variant calling pipelines mentioned above has been released starting with Data Release 42, replacing the previous WGS variant calling pipelines used by the GDC since DR27. The earlier workflow is based on a workflow generated by the Sanger Institute as the following:

- CaVEMan: Single nucleotide variants, which are available in VCF format.
- Pindel: Small indel variants, which are available in VCF format.
- BRASS: Structural variants, which are available in BEDPE format.
- AscatNGS: Copy number variants, which are available as copy number estimates or copy number segment files. Data may be available in tab separated values (.TSV) or plain text file (.TXT)

GATK4 MuTect2 CLI

The GATK4 MuTect2 pipeline follows the documentation from the Broad Institute's best practices.

- GDC Reference Files
- PoN: gatk4_mutect2_4136_pon.vcf.tar
- Reference Seq: GRCh38.d1.vd1 Reference Sequence
- Common variant reference: af-only-gnomad-common-biallelic.grch38.main.vcf.gz

- Germline - resource: af-only-gnomad.hg38.vcf.gz
- Reference image: GRCh38.d1.vd1.fa.img
  - Created by:

```
gatk BwaMemIndexImageCreator
 -I reference.fasta
 -O reference.fasta.img
```

Step 1: Calculate Contamination

Shell

```
##GetPileupSummaries
/usr/local/bin/gatk
--java-options "-XX:+UseSerialGC -Xmx$(java_heap)"
GetPileupSummaries
-R $(reference_seq)
-I $(bam_file) # (For tumor and normal runs individually)
-V $(common_variant_reference)
-L $(intervals) (GDC uses chromosomal level intervals.)
-O $(output).pileups.table

##CalculateContamination
/usr/local/bin/gatk
--java-options "-XX:+UseSerialGC -Xmx$(java_heap)"
CalculateContamination
-I $(tumor_pileups)
-matched $(normal_pileups)
-O $(output_prefix).contamination.table
--tumor-segmentation $(inputs.output_prefix).segments.table
```

Step 2: MuTect2 Calling

If mean read length is greater than or equal to 70bp:

Shell

```
##Mutect2
/usr/local/bin/gatk
--java-options "-XX:+UseSerialGC -Xmx$(java_heap)"
-I $(normal)
-I $(tumor)
-R $(Reference seq)
-normal-sample $(normal sample name)
-pon $(PoN)
--germline-resource $(Germline resource)
-L $(intervals)
-O $(output name)
--bam-output
--f1r2-tar-gz
```

Step 3: Filter MuTect2

Shell

```
##FilterMutectCalls
/usr/local/bin/gatk
--java-options "-XX:+UseSerialGC -Xmx$(java_heap)"
FilterMutectCalls
-V $(unfiltered_vcf)
-R $(Reference Seq)
-O $(output_prefix).gatk4_mutect2.filtered.vcf.gz
--contamination-table $(contamination_table)
--tumor-segmentation $(tumor_segments_table)
--ob-priors $(artifacts_priors)
-stats $(mutect2_stats)
--filtering-stats filtering.stats
```

Step 4: Filter Alignment Artifacts

Shell

```
##GatherBamFiles
/usr/local/bin/gatk
--java-options "-XX:+UseSerialGC -Xmx$(java_heap)"
GatherBamFiles
-O $(output_prefix).unsorted.out.bam
-R $(Reference seq)

##FilterAlignmentArtifacts
/usr/local/bin/gatk
--java-options "-XX:+UseSerialGC -Xmx$(java_heap)"
FilterAlignmentArtifacts
-V $(input_vcf)
-I $(reassembly_bam)
--bwa-mem-index-image $(Reference image)
-R $(Reference seq)
-O $(output_prefix).gatk4_mutect2.raw_filtered.vcf.gz
```

SvABA CLI

SvABA (structural variation and indel analysis by assembly) detects variants by genome - wide local assembly.

Shell

```
svaba run --override-reference-check
-D dbsnp_144.hg38.vcf
-t tumor.bam
-n normal.bam
-k intervals.bed
-v 0
-G GRCh38.d1.vd1.fa (with the following secondary files)
  - .amb
  - .ann
  - .bwt
  - .fai
  - .pac
  - .sa
-p 8
-a <sample_id>
```

Strelka-Manta CLI:

The Manta tool is run first to produce a candidate indel file to have more accurate indel calls and a structural variant file.

Shell

```shell
python manta/bin/configManta.py
--normalBam normal.bam
--tumorBam tumor.bam
--exome
--referenceFasta GRCh38.d1.vd1.fa
--runDir <output_directory> && <output_directory> /runWorkflow.py -m local -j 4
```

The Strelka tool is then run with the Manta candidate indel file as an input parameter. Note that index files are required and need to be in the same directory.

Shell

```shell
/bin/strelka-tool somatic
--normalBam normal.bam
--tumorBam tumor.bam
--referenceFasta GRCh38.d1.vd1.fa
--indelCandidates candidateSmallIndels.vcf.gz
--runDir <output_directory> && <output_directory>/runWorkflow.py -m local -j 4
```

BEDPE File Format

The BEDPE file format, (**b**rowser **e**xtensible **d**ata **p**aired - **e**nd) is designed to concisely describe disjoint genome features, such as structural variations or paired - end sequence alignments. It's an enhanced version of the BED format, as BED does not allow inter - chromosomal feature definitions. In addition, BED only has one strand field, which is insufficient for paired - end sequence alignments, especially when studying structural variation. The BEDPE format is described below.

- **chr*x* (required):** The name of the chromosome on which the *x*th end of the feature exists (x is 1 or 2). Any string can be used. For example, "chr1", "III", "myChrom", "contig1112.23" (use "." for unknown).
- **start*x* (required):** The zero - based starting position of the **first** end of the feature on chr*x*. The first base in a chromosome is numbered 0. The start position in each BEDPE feature is therefore interpreted to be 1 greater than the start position listed in the feature (use -1 for unknown).
- **end*x* (required):** The one - based ending position of the first end of the feature on chr*x*. The end position in each BEDPE feature is one - based (use -1 for unknown).
- **name (optional):** Defines the name of the BEDPE feature. Any string can be used.
- **score (optional):** A score between 0 and 1000.
- **strand*x* (optional):** Defines the strand for the *x*th end of the feature. Either "." (unknown), "+", or "-".

In addition to the above fields, bedtools allows for the addition of user - defined fields to the normal, 10 - column BEDPE format as necessary. These columns are merely "passed through" pairToBed and pairToPair and are not part of any analysis. One would use these additional columns to add extra information (e.g., edit distance for each end of an alignment, or "deletion", "inversion", etc.) to each BEDPE feature.

GATK4 CNV CLI

The GATK4 Copy Number Variant (CNV) pipeline provides a comprehensive framework for analyzing somatic CNVs in genomic data. It begins with CollectReadCounts, which calculates read counts at specified genomic intervals, and DenoiseReadCounts, which uses a Panel of Normals (PoN) generated by GATK's CreateReadCountPanelOfNormals to standardize and denoise these counts. The CollectAllelicCounts gathers reference and alternate allele counts at specified sites.

The ModelSegments calculates copy ratios from denoised copy ratios and segmented minor - allele fractions from allelic counts. These segments are then classified by CallCopyRatioSegments as amplified, deleted, or neutral.

For visualization, PlotDenoisedCopyRatios and PlotModeledSegments generate detailed plots of denoised copy ratios, segmented copy ratios, and minor - allele - fraction estimates.

GATK4 version documentation can be found on the Broad Institute's webpage. The pipeline parameters and steps are described below.

- Inputs: aligned reads files BAMs
- References:
  - intervals.preprocessed.interval_list
  - reference .fa GRCh38.d1.vd1.fa
  - panel of normals

Shell

```
#CollectCounts Tumor/Normal
gatk --java-options -Xmx31500m
CollectReadCounts
--interval-merging-rule OVERLAPPING_ONLY
--output <output_filename>
--input  <inputs.bam>
-L intervals.preprocessed.interval_list
--reference GRCh38.d1.vd1.fa

#CollectAllelicCounts Tumor/Normal
gatk --java-options -Xmx31000m
CollectAllelicCounts
--output <output_filename>
--input <inputs.bam>
-L <common_sites_hg38_lifted.list>
--minimum-base-quality 20
--reference <GRCh38.d1.vd1.fa>

#DenoiseReadCounts Tumor/Normal
gatk --java-options -Xmx31000m
DenoiseReadCounts
--standardized-copy-ratios <output_filename.standardizedCR.tsv>
--denoised-copy-ratios <output_filename.denoisedCR.tsv>
--count-panel-of-normals <read_count_panel_of_normal.hdf5>
--input <CollectCounts.hdf5>

#ModelSegments Tumor/Normal
adjust "--minimum-total-allele-count-case" threshold if ModelSegments cannot find enough reads or f
ind extra noises.
```

```
gatk --java-options -Xmx29000m ModelSegments
--allelic-counts <CollectAllelicCounts.tsv>
--denoised-copy-ratios <DenoiseReadCounts.denoisedCR.tsv>
--output-prefix _wgs_gdc_realn
--genotyping-base-error-rate 0.05
--genotyping-homozygous-log-ratio-threshold -10.0
--kernel-approximation-dimension 100
--kernel-scaling-allele-fraction 1.0
--kernel-variance-allele-fraction 0.025
--kernel-variance-copy-ratio 0.0
--maximum-number-of-segments-per-chromosome 1000
--maximum-number-of-smoothing-iterations 10
--minimum-total-allele-count-case 0
--minimum-total-allele-count-normal 0
--minor-allele-fraction-prior-alpha 25.0
--normal-allelic-counts <CollectAllelicCounts.tsv>
--number-of-burn-in-samples-allele-fraction 50
--number-of-burn-in-samples-copy-ratio 50
--number-of-changepoints-penalty-factor 1.0
--number-of-samples-allele-fraction 100
--number-of-samples-copy-ratio 100
--number-of-smoothing-iterations-per-fit 0
--output <outputs.seg>
--smoothing-credible-interval-threshold-allele-fraction 2.0
--smoothing-credible-interval-threshold-copy-ratio 2.0
--window-size 8
--window-size 16
--window-size 32
--window-size 64
--window-size 128
--window-size 256

#CallCopyRatioSegments Tumor/Normal
gatk --java-options -Xmx31000m CallCopyRatioSegments
--output <output_filename>
--calling-copy-ratio-z-score-threshold 2.0
--input <ModelSegments.seg>
--neutral-segment-copy-ratio-lower-bound 0.9
--neutral-segment-copy-ratio-upper-bound 1.1
--outlier-neutral-segment-copy-ratio-z-score-threshold 2.0

#PlotDenoisedCopyRatios Tumor/Normal
gatk --java-options -Xmx31000m PlotDenoisedCopyRatios
--denoised-copy-ratios <DenoiseReadCounts.denoisedCR.tsv>
--output-prefix <output_filename>
--minimum-contig-length 1000000
--output out
--sequence-dictionary GRCh38.d1.vd1.dict
--standardized-copy-ratios <DenoiseReadCounts.standardizedCR.tsv>

#PlotModeledSegments Tumor/Normal
gatk --java-options -Xmx31000m PlotModeledSegments
--denoised-copy-ratios <DenoiseReadCounts.denoisedCR.tsv>
--output-prefix <output_filename>
--allelic-counts <CollectAllelicCounts.output>
--minimum-contig-length 1000000
--segments <ModelSegments.seg>
--output <output_filename.seg>
--sequence-dictionary GRCh38.d1.vd1.dict
```

CNV from WGS File Format

As of Data Release 42, copy number segment data from the GATK4 pipeline is available for download. See below for a description of the available columns in each file.

- **GDC_Aliquot_ID**: The GDC submitter ID for the aliquot collected from the sample
- **Start**: Starting position of the segment
- **End**: Ending position of the segment
- **Segment_Mean**: This is equal to log2(copy-number/ 2) for the segment
- **Num_Probes**: Legacy field from when segment files were derived from SNP 6.0 arrays. This has no significance for NGS-derived CNV data

AscatNGS (**now deprecated at GDC**), originally developed by Raine *et al* (2016) (GitHub page), indicates the DNA copy number changes affecting a tumor genome when comparing to a matched normal sample. See below for a description of the copy number segment and copy number estimation files produced by AscatNGS:

- GDC Aliquot: The GDC ID for the aliquot collected from the sample (copy number segment files only).
- Gene ID: The gene ENSEMBL ID (copy number variant only).
- Gene Name: The gene symbol (copy number variant only).
- Chromosome: The name of the chromosome on which the copy number change exists.
- Start: The starting position of the copy.
- End: The ending position of the copy.
- Copy Number: The weighted median of the strand copy numbers [9].
- Major Copy Number: The greater strand copy number of the two strands of the DNA (copy number segment files only).
- Minor Copy number: The smaller strand copy number of the two strands of the DNA (copy number segment files only).
- Max. Copy number: The highest copy number for overlapped segment (copy number variant only).
- Min. Copy number: The lowest copy number for overlapped segment (copy number variant only).