Whole-Genome Small Variant Calling

Table of contents

This section explores whole-genome germline small variant calling using the HG002 Genome-in-a-Bottle Sample.

Software Requirements

- samtools
- BWA
- Parabricks version 4.0 or later

Hardware Requirements

These are the minimum hardware requirements for this how-to:

- 512GB of disk space (preferably on "balanced" or SSD storage)
- Two Nvidia V100 GPUs; the commands and time estimates below will utilize four Nvidia V100 GPUs
- 24 CPU cores; the commands below use 32 vCPU cores
- 48GB of CPU RAM per GPU

The examples below use a Google Cloud Project VM with 32 vCPU cores, 120 GB of RAM, and four NVIDIA V100 GPUs running Ubuntu 20.04.

Introduction

This how-to will run through a full whole-genome germline pipeline for calling SNPs, MNPs, and indels on real 30X short-read human data. Such analyses are common in a variety of settings:

- Population studies
- Genome-wide association studies
- Trio analysis (when combined with downstream filtering)

- When analyzing data from biobanks (such as reanalysis of 1000 Genomes data)
- When looking for possible hereditary cancer predisposition mutations (e.g. Lynch Syndrome or mutations in certain BRCA genes)
- When looking for disease-associated mutations in clinical sequencing

The data was generated from the son in a trio sequenced by the Genome In A Bottle Consortium. This sample, identified as HG002, has been highly characterized across multiple sequencing platforms and variant callers, and a high-quality "truth set" of variants exists that allows you to check the results.

After variant calling you'll want to annotate the VCF with one or more databases to determine which variants are common or are associated with disease (such as those observed frequently in 1000 Genomes), filter out those common variants and then use the NVIDIA Parabricks tools for quality control to assess the variant caller results. We don't cover annotation or filtering in this How-To.

The first steps of this workflow (alignment, variant calling, and quality control) are common across many different analyses. Depending on your use case, however, the annotation and filtering steps may differ.

Example data

The example data for this how-to resides in a public Google Cloud Storage bucket. It can be downloaded using the `gsutil` tool from the Google Cloud SDK; alternatively, the file can be downloaded with `wget` using the second set of instructions below. Note that there are two fastq files to download (ending in "R1.fastq.gz" and "R2.fastq.gz").

```
$ gsutil cp gs://brain-genomics-public/research/sequencing/fastq/hiseqx/wgs_pcr_free/30x/HG002.hiseqx.pcr-free.30x.R1.fastq.gz . $ gsutil cp gs://brain-genomics-public/research/sequencing/fastq/hiseqx/wgs_pcr_free/30x/HG002.hiseqx.pcr-free.30x.R2.fastq.gz .
```

To download the data with wget (if gsutil is not available):

```
$ wget https://storage.googleapis.com/brain-genomics-public/research/sequencing/fastq/hiseqx/wgs_pcr_free/30x/HG002.hiseqx.pcr-free.30x.R1.fastq.gz $ wget https://storage.googleapis.com/brain-genomics-public/research/sequencing/fastq/hiseqx/wgs_pcr_free/30x/HG002.hiseqx.pcr-free.30x.R2.fastq.gz
```

If you have your own data, you'll likely be able to substitute it directly into the commands in this how-to to get accurate, multi-caller variant calls.

Downloading and Indexing a Reference Genome and Known Sites

Next, you will download and index a reference genome to use in alignment and variant calling. You will use GRCh38 without alt contigs. Indexing should take between 30 minutes and one hour.

```
$ wget ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/GCA_000001405.15_GRCh38/ ## Unzip and index the reference $ gunzip GCA_000001405.15_GRCh38_no_alt_analysis_set.fna.gz ## Create an FAI index $ samtools faidx GCA_000001405.15_GRCh38_no_alt_analysis_set.fna ## Create the BWA indices $ bwa index GCA_000001405.15_GRCh38_no_alt_analysis_set.fna
```

To generate a BQSR report (used for improving the base qualities within the BAM and downstream HaplotypeCaller calls), we'll also need a VCF file with known variant calls. We'll retrieve these from the Broad HG38 resource bundle:

```
$ wget https://storage.googleapis.com/genomics-public-data/resources/broad/hg38/v0/Mills_and_1000G_gold_standard.indels.hg38.vcf.gz $ wget https://storage.googleapis.com/genomics-public-data/resources/broad/hg38/v0/Mills_and_1000G_gold_standard.indels.hg38.vcf.gz.tbi
```

Aligning Reads

The fq2bam (BWA-MEM + GATK) command runs read alignment -- as well as sorting, duplicate marking, and base quality score recalibration (BQSR) -- according to GATK best practices, but at a much faster rate than community tools by leveraging up to 8 NVIDIA GPUs.

Note that this how-to adds a custom read group to make the sample easier to identify if it is later merged with others in downstream analysis; this is essential when running alignment for matched tumor-normal or multi-sample analyses. You also have to pass a set of known sites to generate the BQSR report.

```
$ RGTAG="@RG\tID:HG002\tLB:lib\tPL:Illumina\tSM:HG002\tPU:HG002" # This command assumes all the inputs are in INPUT_DIR and all the outputs go to OUTPUT_DIR. docker run --rm --gpus all --volume INPUT_DIR:/workdir --volume OUTPUT_DIR:/outputdir \ \ --gpus all \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bam \ --ref /workdir/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna \ --in-fq /workdir/HG002.hiseqx.pcr-free.30x.R1.fastq.gz /workdir/HG002.hiseqx.pcr-free.30x.R2.fastq.gz "${RGTAG}" \ --knownSites /workdir/Mills_and_1000G_gold_standard.indels.hg38.vcf.gz \ --out-bam /outputdir/HG002.hiseqx.pcr-free.30x.pb.bam \ --out-recal-file /outputdir/HG002.hiseqx.pcr-free.30x.pb.BQSR-report.txt
```

The read alignment, sorting, duplicate marking and indexing process (all included in fq2bam (BWA-MEM + GATK)) should take about 50 minutes with four NVIDIA V100 GPUs. The output of fq2bam (BWA-MEM + GATK) is a BAM file, a BAI index, and a BQSR report.

```
$ ls HG002.hiseqx.pcr-free.30x.pb* HG002.hiseqx.pcr-free.30x.pb.bam HG002.hiseqx.pcr-free.30x.pb.bam.BAI HG002.hiseqx.pcr-free.30x.pb.BQSR-report.txt
```

These files will be used as inputs to the multiple SNP/indel callers run in the next step.

Generating a Comprehensive Set of Variant Calls with DeepVariant and HaplotypeCaller

Parabricks currently supports several variant callers (DeepVariant, HaplotypeCaller, Mutect2). Each of these uses slightly different methods for calling variants, with particular trade-offs in sensitivity and specificity for each.

This how-to runs all three callers to generate a comprehensive set of variant calls. If you were looking for variants of clinical interest, you might consider taking the intersection of two or more callers to generate a highly specific callset. On the other hand, if you weren't concerned about false positives but needed a highly-sensitive callset, you could take the union of all three callers. Such vote-based consensus approaches have been employed in numerous large studies to address the shortcomings of individual callers.

First, run DeepVariant, a deep-learning based variant caller originally developed by Google; Parabricks provides an accelerated version of DeepVariant that is 10-15X faster than the community version. DeepVariant should run in under 30 minutes on a 4xV100 system.

```
$ docker run \ --gpus all \ --workdir /workdir \ --rm \ --volume $(pwd):/workdir \ --volume $(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun deepvariant \ --in-bam /workdir/HG002.hiseqx.pcr-free.30x.pb.bam \ --ref /workdir/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna \ --out-variants /outputdir/HG002.hiseqx.pcr-free.30x.pb.deepvariant.vcf
```

Next, run HaplotypeCaller, long considered the gold standard of germline small variant callers. Parabricks HaplotypeCaller takes roughly 15 minutes on four NVIDIA V100 GPUs, a speed increase of roughly 60X over the community version.

```
$ docker run \ --gpus all \ --workdir /workdir \ --rm \ --volume $(pwd):/workdir \ --volume $(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun haplotypecaller \ --in-bam /workdir/HG002.hiseqx.pcr-free.30x.pb.bam \ --in-recal-file /workdir/HG002.hiseqx.pcr-free.30x.pb.BQSR-report.txt \ --ref /workdir/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna \ --out-variants /outputdir/HG002.hiseqx.pcr-free.30x.pb.haplotypecaller.vcf
```

The results of DeepVariant and HaplotypeCaller are plain-text VCF files.