DeepVariant is an analysis pipeline that uses a deep neural network to call genetic variants from next-generation DNA sequencing data.

This page explains how to run DeepVariant (https://github.com/google/deepvariant) on Google Cloud using a single Compute Engine instance.

There are more complex configurations available in the DeepVariant GitHub repository (https://github.com/google/deepvariant#external-solutions). For example, you can run DeepVariant using multiple instances. These variations provide improvements in processing speed and reduced costs.

Running DeepVariant consists of three stages:

1. **Making examples**: DeepVariant pre-processes the input data and saves examples from the data using an internal TensorFlow format. You can run this stage in parallel where all of the input shards are processed independently.
2. **Calling variants**: DeepVariant runs a deep neural network that makes inferences from the examples and saves them into shared files using an internal TensorFlow format.
3. **Post-processing variants**: DeepVariant converts variants from the internal TensorFlow format to VCF or gVCF files. This stage runs on a single thread.

In this tutorial, you run all of these stages using a single instance. The first and second stages (making examples and calling variants, respectively) can benefit from parallelization on multiple cores, but the third stage (post-processing variants) does not have the same benefits because it runs on a single thread.

After completing this tutorial, you'll know how to:

- Run DeepVariant on Google Cloud

This tutorial uses billable components of Google Cloud, including:

- Compute Engine

Use the Pricing Calculator (/products/calculator) to generate a cost estimate based on your projected usage. New Cloud Platform users might be eligible for a free trial (/free-trial).

1. Sign in (https://accounts.google.com/Login) to your Google Account.

   If you don't already have one, sign up for a new account (https://accounts.google.com/SignUp).
2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

Go to the project selector page (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (/billing/docs/how-to/modify-project).
4. Enable the Compute Engine API.

   Enable the API (https://console.cloud.google.com/flows/enableapi?apiid=compute.googleapis.com&redirect=https://console.cloud.google.com)
5. Install and initialize the Cloud SDK (/sdk/docs/).

★ **Tip**: Need a command prompt? You can use the Google Cloud Shell (https://console.cloud.google.com?cloudshell=true). The Google Cloud Shell is a command line environment that already includes the Google Cloud SDK, so you don't need to install it.

Before running DeepVariant, you need to create a Compute Engine instance on which DeepVariant will run. You can create a Linux virtual machine instance in Compute Engine using either the Google Cloud Console or the `gcloud` command-line tool.

Allow a short time for the instance to start up. Once ready, it will be listed on the VM Instances (https://console.cloud.google.com/compute/instances) page with a green status icon.

You can connect to the instance using either the Cloud Console or the `gcloud` tool:

Configure your environment and run DeepVariant by completing the following steps on the Compute Engine instance you created:

1. Install Docker Community Edition (CE). (https://docs.docker.com/install/) by running the following commands:

2. Configure the DeepVariant environment variables by copying and pasting the following commands:

3. Create the local directory structure for the input data directory and the output directory:

4. This tutorial uses a publicly available HG002 genome at 30x coverage mapped to GRCh37 reference. However, to ensure a quicker runtime, you will add the `--regions 20` flag when you run DeepVariant so that DeepVariant only runs on chromosome 20 (chr20).

   The sample data was created using Illumina sequencing, but DeepVariant also supports the following other types of input data:

   - Whole genome (Illumina) (WGS)
   - Exome (Illumina) (WES)
   - Whole genome (PacBio)

   Using the `gsutil cp` (/storage/docs/gsutil/commands/cp) command, copy the input test data from the `deepvariant` Cloud Storage bucket to the directories on the instance you created in the previous step:

5. DeepVariant is a containerized application staged in a pre-built Docker image in Container Registry (/container-registry/docs/pushing-and-pulling). To pull the image, run the following command:

6. Run the following command to start DeepVariant. The total running time for the command is roughly eight minutes.

The following table describes the flags passed in to the command:

Flag	Description
`model_type`	DeepVariant supports several different types of input data. This tutorial uses Whole Genome Sequencing (WSG).
`ref`	The location of the reference FASTA file.
`reads`	The location of the input BAM file.
`output_vcf`	The location of the output VCF files.
`output_gvcf`	The location of the output gVCF files.
`regions`	(Optional) A space-separated list of chromosome regions to process. Individual elements can be region literals, such as `chr20:10-20` or paths to BED (https://bedtools.readthedocs.io/en/latest/content/general-usage.html#bed-format)/BEDPE (https://bedtools.readthedocs.io/en/latest/content/general-usage.html#bedpe-format) files.
`num_shards`	The number of shards to run in parallel. For best results, set the value of this flag to the number of cores on the machine where DeepVariant runs.

If the command starts to run successfully, it outputs a message similar to the following:

7. After DeepVariant finishes, it outputs the following files to the `deepvariant-run/output` directory:
    - `HG002.output.g.vcf.gz`
    - `HG002.output.g.vcf.gz.tbi`
    - `HG002.output.vcf.gz`
    - `HG002.output.vcf.gz.tbi`

    Run the following command to list the files in the output directory, and check that all of the output files display:

The following table shows the approximate runtime and costs when running DeepVariant using a 30x whole genome sample in a BAM file. These estimates do not include the time required to set up the instance and download any sample data from Cloud Storage.

The table contains estimates for preemptible VMs (/preemptible-vms/) and non-preemptible VMs. The runtime estimates are based on using non-preemptible VMs.

Preemptible VMs are up to 80% cheaper than regular VMs. However, Compute Engine might terminate (preempt) these instances if it requires access to those resources for other tasks. Additionally, preemptible VMs are not covered by any Service Level Agreement (SLA), so if you require guarantees on turnaround time, do not use the `--preemptible` flag.

See the Compute Engine best practices (/compute/docs/instances/create-start-preemptible-instance#best_practices) for how to effectively use preemptible VMs.

Machine type	Runtime in hours	Cost (non-preemptible)	Cost (preemptible)
n1-standard-8	19.7	$8.06	$2.16
n1-standard-16	11.3	$8.91	$2.14
n1-standard-32	7.25	$11.20	$2.54
n1-standard-64	5.65	$17.30	$3.78
n1-standard-96	4.52	$20.80	$4.48

To avoid incurring charges to your Google Cloud account for the resources used in this tutorial:

After you've finished the Running DeepVariant tutorial, you can clean up the resources you created on Google Cloud so you won't be billed for them in the future. The following sections describe how to delete or turn off these resources.

The easiest way to eliminate billing is to delete the project you used for the tutorial.

To delete the project:

ıg: Deleting a project has the following consequences:

If you used an existing project, you'll also delete any other work you've done in the project.

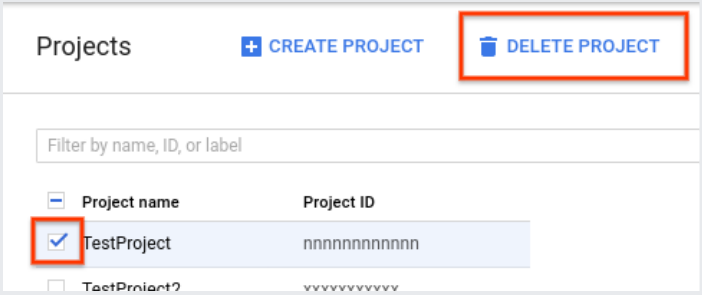
You can't reuse the project ID of a deleted project. If you created a custom project ID that you plan to use in the future, delete the resources inside the proj instead. This step ensures that URLs that use the project ID, such as an **`appspot.com`** URL, remain available.

are exploring multiple tutorials and quickstarts, reusing projects instead of deleting them prevents you from exceeding project quota limits.

1. In the Cloud Console, go to the Projects page.

Go to the Projects page (https://console.cloud.google.com/iam-admin/projects)

2. In the project list, select the project you want to delete and click **Delete project**.



3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

- Read the DeepVariant documentation (https://github.com/google/deepvariant) on GitHub.
- Read a Google AI blog post (https://ai.googleblog.com/2017/12/deepvariant-highly-accurate-genomes.html) about DeepVariant's open source release.
- If you have DeepVariant questions, send an email to the Google Cloud and Google Brain teams at google-genomics-contact@google.com (mailto:google-genomics-contact@google.com).