Welcome to NVIDIA Parabricks v4.3.1

Table of contents

What's New? 15

4.3.1-1 Release Notes 15

4.3.0-1 Release Notes 18

4.2.1-1 Release Notes 21

4.2.0-1 Release Notes 24

4.1.1-1 Release Notes 26

4.1.0-1 Release Notes 26

4.0.0-1 Release Notes 27

Getting Started with NVIDIA Parabricks 30

Software Overview 37

Software Tools 39

NVIDIA Parabricks WDL/Nextflow Workflows 40

Compatible CPU Software Versions 41

Best Performance 42

Tutorials 49

Getting The Sample Data 50

FQ2BAM Tutorial 51

HaplotypeCaller Tutorial 54

Cloud Usage Guides 57

DeepVariant training using Parabricks 58

How-Tos 72

Whole-Genome Small Variant Calling 72

Whole-Genome Somatic Small Variant Calling 77

Tool Reference 87

applybqsr 88

bam2fq 92

bammetrics 95

bamsort 98

bqsr 103

collectmultiplemetrics 106

dbsnp 109

deepsomatic 112

deepvariant 116

deepvariant_germline 126

fq2bam (BWA-MEM + GATK) 140

fq2bam_meth 149

fq2bamfast (BWA-MEM + GATK) 160

genotypegvcf 169

germline (GATK Germline Pipeline) 172

haplotypecaller 187

indexgvcf 197

markdup 199

minimap2 (Beta) 203

mutectcaller 210

pacbio_germline (Beta) 219

postpon 232

prepon 234

rna_fq2bam 236

somatic (Somatic Variant Caller) 246

starfusion 258

Grace Hopper Superchip 261

Help 263

References 265

Publications 265

Third Party Software Notices 266

Comparison with Baseline Tools 267

End User License Agreements 268

List of Figures

Figure 0.

Figure 1. Image Instances

Figure 2. Image Launch Instances

Figure 3. Image Name And Tags

Figure 4. Image App And Os Images

Figure 5. Image Instance Type

Figure 6. Image Key Pair

Figure 7. Image Configure Storage

Figure 8. Image Summary

Figure 9. Image Instances Info

Figure 10. Image Connect To Instance

Figure 11. Image Install Parabricks

Figure 12. Image Parabricks On Ngc

Figure 13. Image Docker Pull Cmd

Figure 14. Image Tree Cmd

Figure 15. Image Banner

Figure 16. Image Ready2run

Figure 17. Image Create Workflow Run

Figure 18. Image Run Details

Figure 19. Image Runs

Figure 20. Image Result

Figure 21. Image Starting A Compute Instance

Figure 22. Image Create Pb Resource

Figure 23. Image Set Instance Details

Figure 24. Image Select Image

Figure 25. Image Choose Size

Figure 26. Image Admin Acct

Figure 27. Image Validation Passed

Figure 28. Image Go To Resource

Figure 29. Image Connect

Figure 30. Image Ngc

Figure 31. Image Docker Pull

Figure 32. Image Wget Sample Data

Figure 33. Image Untar

Figure 34. Image Output

Figure 35. Image Homepage

Figure 36. Image Tool List

Figure 37. Image Fq2bam Tool

Figure 38. Image Inputs And Outputs

Figure 39. Image Options

Figure 40. Image Interval Help

Figure 41. Image Select File

Figure 42. Image View Log

Figure 43. Image Terminal

Figure 44. Image View Inputs And Outputs

Figure 45. Image View Project Id

Figure 46. Image Vm Instances

Figure 47. Image Instance Settings

Figure 48. Image Machine Configuration

Figure 49. Image Boot Disk

Figure 50. Image Create

Figure 51. Image Ssh

Figure 52. Image Install Drivers

Figure 53. Image Docker Ps

Figure 54. Image Ngc1

Figure 55. Image Docker Pull1

Figure 56. Image Tree

Figure 57. Image Startup

Figure 58. Image File List

Figure 59. Available Modules

Figure 60. Running On Nf Core

Figure 61. ControlConsoleGUI

Figure 62. Marketplaces

Figure 63. NvidiaApps

Figure 64. NvidiaGpuCloud

Figure 65. CreateComputeInstance1

Figure 66. CreateComputeInstance2

Figure 67. CreateComputeInstance3

Figure 68. Icon

Figure 69. CreateComputeInstance4

Figure 70. SshCommand

Figure 71. WhichCommand

Figure 72. LsblkCommand

Figure 73. DockerPullCommand

Figure 74. TarCommand

Figure 75. ProgressMeter

Figure 76. Output

Figure 77. Image Workspace

Figure 78. Image Clara Workspace

Figure 79. Image Select Data

Figure 80. Image Status

Figure 81. Image Execution Directory

Figure 82. Image Output Files

Figure 83. HowTo SomaticCallingPlot

Figure 84. Fq2bam

Figure 85. Fq2bam Meth Diagram

Figure 86. Fq2bam

Figure 87. Germline

List of Tables

Table 0.

Table 1.

- What's New?
- Getting Started with NVIDIA Parabricks
- Software Overview
- Best Performance
- Tutorials
- How-Tos
- Tool Reference
- Grace Hopper Superchip
- Help
- References

How the Documentation is Organized

- This page contains a brief overview of NVIDIA Parabricks: What it is, what it can do, and how to use it.

- What's New? covers what's changed since the previous release: new tools, improvements to existing tools, and bug fixes.

- Getting Started with NVIDIA Parabricks focuses on all the steps of setting up the software, including requirements, examples and optimizing it for performance

- Software Overview is a more detailed discussion of the Parabricks tools, how to use Parabricks in a WDL or Nextflow environment, and its compatibility with comparable CPU versions of the software.

- Best Performance gives tips on achieving optimal performance with the Parabricks software suite.

- Tutorials walks you through a single use of Parabricks using an example dataset. The steps will familiarize the users with the software and walk you through a reproducible example. It will start from a reference and FASTQ files to a BAM file, then do variant calling on the BAM file, and produce a VCF file.

- How-Tos explores larger, more involved tasks, examining a wider variety of options, tools, and workflows. Owing to the larger data sets in use, a more capable hardware platform may be required (more GPUs, more memory, etc).

- Tool Reference contains reference documentation for each tool, organized both by category and alphabetically by tool name. It also tells users how to compare the output of Parabricks with the output from the baseline tools. A list of publications referencing Parabricks, a list of frequently asked questions, and pointers on getting more help and information are also part of this section

- Grace Hopper Superchip contains guides and references for running Parabricks on the new Grace Hopper Superchip.

What is Parabricks?

Parabricks is a free software suite for performing secondary analysis of next generation sequencing (NGS) DNA and RNA data. It delivers results at blazing fast speeds and low cost. Parabricks can analyze 30x WGS (whole human genome) data in about 10 minutes, instead of 30 hours for other methods. Its output matches commonly used software, making it fairly simple to verify the accuracy of the output.

Why use Parabricks?

Under the hood, Parabricks achieves this performance through tight integration with GPUs, which excel at performing data-parallel computation much more effectively than traditional CPU-based solutions. Parabricks was built from the ground up by GPU computing and Deep Learning experts who wanted to develop the fastest and most efficient possible implementation of common genomics algorithms used in secondary analysis.

Learn more at the Parabricks developer page.

How can I get Parabricks?

Parabricks is freely available as a public container on NGC for use on-premises or any cloud service platforms and providers. You can learn more about Parabricks on our webpage, including how to purchase enterprise support for Parabricks through NVIDIA AI Enterprise with guaranteed response times, priority security notifications and access to AI experts from NVIDIA. Users on DGX Cloud are able to utilize NVIDIA AI Enterprise for free.

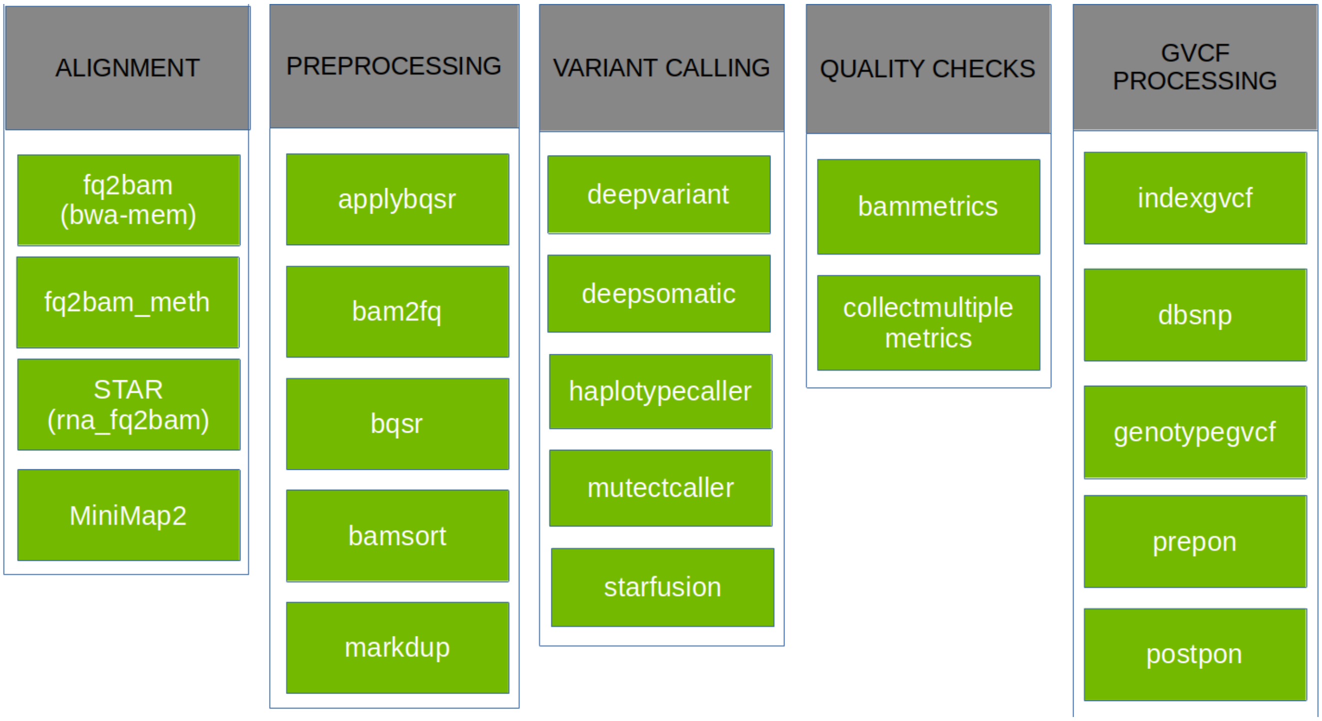
See the following Cloud Startup guides for more information on using Parabricks in the cloud:

- AWS
- Azure
- DNAnexus
- GCP
- nf-core
- OCI
- Terra

Software Overview

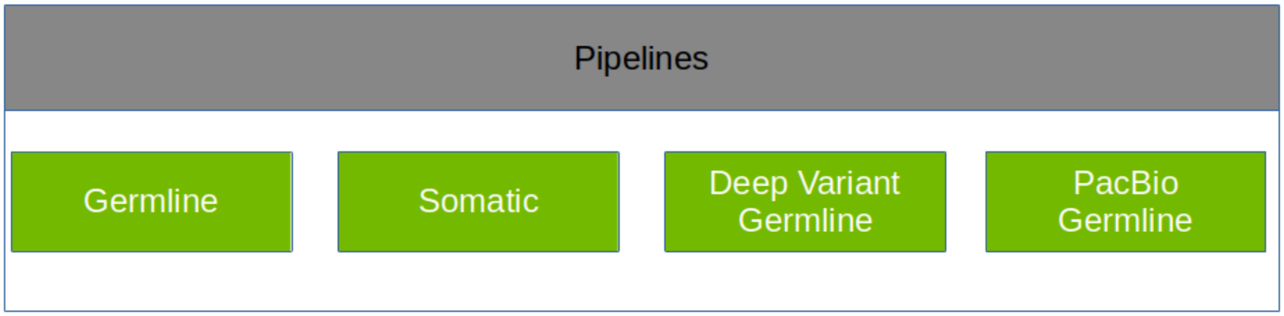
Parabricks is a software suite for genomic analysis. It delivers major improvements in throughput time for common analytical tasks in genomics, including germline and somatic analysis. The core of the Parabricks software is its tight integration with the GPU, which takes raw data and transforms it according to the user's requirements.

Parabricks supports the tools shown below:



ALIGNMENT	PREPROCESSING	VARIANT CALLING	QUALITY CHECKS	GVCF PROCESSING
fq2bam (bwa-mem)	applybqsr	deepvariant	bammetrics	indexgvcf
fq2bam_meth	bam2fq	deepsomatic	collectmultiple metrics	dbsnp
STAR (rna_fq2bam)	bqsr	haplotypecaller		genotypegvcf
MiniMap2	bamsort	mutectcaller		prepon
	markdup	starfusion		postpon

Additionally, the Parabricks tool suite provides a number of variant calling pipelines that are each a combination of several individual tools, combining into one tool what would otherwise be a multi-step process. These pipelines are:



Pipelines			
Germline	Somatic	Deep Variant Germline	PacBio Germline

Parabricks has been tested on Dell, HPE, IBM, and NVIDIA servers at Amazon Web Services, Google Cloud, Oracle Cloud Infrastructure, and Microsoft Azure.

How to Get Help

1. For technical support, updated user guides, and other Clara Parabricks documentation, see the NVIDIA page.

2. Answers to most FAQs can be found on the developer forum.

What's New?

New features, performance improvements and bug fixes by release.

- 4.3.1-1 Release Notes
- 4.3.0-1 Release Notes
- 4.2.1-1 Release Notes
- 4.2.0-1 Release Notes
- 4.1.1-1 Release Notes
- 4.1.0-1 Release Notes
- 4.0.0-1 Release Notes

For further information see the NVIDIA Parabricks datasheet.

4.3.1-1 Release Notes

Highlights:

- New tool, deepsomatic for somatic variant calling.

- The latest Parabricks toolkit (v4.3.1) is now fully supported on Grace Hopper, see Grace Hopper Superchip.

- deepvariant version 1.6.1 updated, new option added.

- haplotypecaller bug fixes and 3 new options added.

- mutectcaller a few bug fixes.

- rna_fq2bam bug fixes and feature added in wrapper.

- minimap2 (Beta) bug fixes, version update, and improved support for ONT data.

- fq2bamfast and fq2bam: Bug fixes and performance improvements. Additional option to monitor approximate CPU utilization and host memory usage during execution (`--monitor-usage`).

New Tools

With Parabricks 4.3.1 we are releasing an accelerated deepsomatic tool. DeepSomatic builds on the deep learning-based variant caller DeepVariant. It processes aligned reads from tumor and normal samples (in BAM or CRAM format), generates pileup image tensors, classifies these tensors using a convolutional neural network, and outputs somatic variants in standard VCF or gVCF files.

Improvements

Tool Updates

haplotypecaller:

- Adds the following new options:
    - `--minimum-mapping-quality`
    - `--mapping-quality-threshold-for-genotyping`
    - `--enable-dynamic-read-disqualification-for-genotyping`
- Improved performance by leveraging AVX512 instructions for CPU-based PairHMM computation.

mutectcaller:

- Improved performance by leveraging AVX512 instructions for CPU-based PairHMM computation.

deepvariant:

- Updates to match the baseline version v1.6.1.
- Adds the new option `--haploid-contigs`
- Improved performance for short-read mode through increased GPU utilization and kernel optimizations.

rna_fq2bam:

- Supports passing `--out-chim-type` multiple times.

fq2bamfast:

- Improved alignment performance on Hopper GPUs through increased use of DPX instructions.
- Improved performance on multi-GPU runs; for example, on DGX H100.
- Improved error detection for improper FASTQ inputs through `--in-fq` or `--in-se-fq`. Previously recorded a utf-8 decode error.
- Additional option to monitor approximate CPU utilization and host memory usage during execution (`--monitor-usage`).

fq2bam:

- Improved error detection for improper FASTQ inputs through `--in-fq` or `--in-se-fq`. Previously recorded a utf-8 decode error.
- Additional option to monitor approximate CPU utilization and host memory usage during execution (`--monitor-usage`).

minimap2:

- Updated map-pbmm2 preset to match the updated versions of minimap2 (v2.26) and pbmm2 (v1.13.0).

Bug Fixes

- mutectcaller and haplotypecaller: Fixed a wrong alignment offset value in smith-waterman algorithm.
- mutectcaller and haplotypecaller: Fixed a crash on GPU when running in low memory mode.
- mutectcaller: Fixed the wrong active probability value when the pileup size is 0.
- mutectcaller: Fixed a max coverage overflow bug.
- rna_fq2bam: Fixed an error when passing "WithinBAM_SoftClip" or "WithinBAM_HardClip" to `--out-chim-type`.
- minimap2 (Beta): Fixed support for Oxford Nanopore Technologies (ONT) data with minimap2.
- fq2bamfast: Fix rare erroneous assertion (`Workspace not big enough, expected desiredSize &lt;= cubWorkspaceSize, exiting`). Case will now be handled correctly and fall back to CPU recovery if needed.
- deepvariant: Fixed bug related to Smith-Waterman computation on CPU.

For further information see the Parabricks datasheet.

4.3.0-1 Release Notes

Highlights:

- New tool, fq2bam_meth for accelerated DNA methylation analysis.

- Germline resource mode and force calling mode are supported in mutectcaller.
- Support for writing CRAM files using queryname-based sorting has been added into bamsort.
- Parabricks toolkit (v4.2) is now fully supported on Grace Hopper, see Grace Hopper Superchip.
- Performance improvements in germline and deepvariant_germline running on DGX H100.
- deepvariant version 1.6 updated.
- minimap2 performance improvements and bug fixes.
- fq2bamfast performance improvements and bug fixes.

New Tools

New tool for bisulfite sequencing data fq2bam_meth based on bwa-meth. Our tool, fq2bam_meth, implements compatible pre- and post-processing around BWA MEM for DNA methylation analysis. It uses the same accelerated alignment code as is used in fq2bamfast to produce fast and accurate alignment.

Improvements

Tool Updates

bamsort:

- Supports CRAM file write on queryname-based sorting. It auto-detects cram file extension on output file.

mutectcaller:

- Adds the following new options:
  - `--mutect-germline-resource`
  - `--mutect-alleles`
  - `--force-call-filtered-alleles`

deepvariant:

- Updates to match the baseline version v1.6.

minimap2 (Beta):

- Reduced reader buffer size to shorten the time it takes to start processing.

fq2bamfast:

- Speed improvements.
- Added support for BWA MEM options: `-B` (values up to 15), `-T`, `-L`, and `-U`.
- Added support for reads longer than 500 bp using CPU recovery mode (note that speed will be slower and memory usage will be higher). Set `--max-read-length` to the desired max read length for the FASTQ filter.

Improvements spanning multiple tools

- Better messaging in filehandle when reading index files to avoid user confusion.
- Better error checking when reading FASTQ files: checks that *each* FASTQ read name line starts with '@'.

Bug Fixes

- mutectcaller and haplotypecaller: Fixed a random segfault in bamOut mode.
- haplotypecaller: Fixed an "allele out of index" bug in gvcf mode.
- haplotypecaller: Fixed an GPU shared memory overflow bug in gvcf mode.
- haplotypecaller, mutectcaller and deepvariant: Fixed a wrong return value 0 when the run fails.
- minimap2 (Beta): Corrected banner name when running minimap2.
- minimap2 (Beta): Fixed overflow issue in postsort.
- minimap2 (Beta): Removed max read size requirement that made some inputs unable to run.
- fq2bamfast: Fix for edge case (related to multiple hits) with `-C` option in BWA MEM to copy auxiliary tags from FASTQ comments.

For further information see the Parabricks datasheet.

4.2.1-1 Release Notes

Two new beta tools have been released: fq2bamfast and minimap2.

A new beta pipeline has been released: pacbio_germline.

Performance improvements include fq2bamfast, a faster version of the existing fq2bam and the ability to run the variant caller in germline and deepvariant_germline in parallel with BAM generation.

A number of bugs have been fixed. Most notably, fq2bam read filtering could have failed with read sizes more than ~500 basepairs.

New Tools

With this release we have added two new tools:

- fq2bamfast
- minimap2

We have also added one new pipeline:

- pacbio_germline

The new beta fq2bamfast is a re-implementation of accelerated BWA-MEM present in fq2bam to better utilize GPUs such as A100 and H100.

Improvements

Tool Updates

mutectcaller:

- Adds the following new options:
  - `--active-probability-threshold`
  - `--genotype-germline-sites`
  - `--genotype-pon-sites`

  - `--initial-tumor-lod`
  - `--mutect-bam-output`
  - `--pruning-lod-threshold`
  - `--tumor-lod-to-emit`
  - `-max-reads-per-alignment-start` (as part of the `--mutectcaller-options` option)

fq2bam and associated pipelines: germline pipeline, deepvariant_pipeline, and somatic pipeline:

- New compression options for `--gpuwrite`: use `--gpuwrite-deflate-algo 3` for more compression at slightly slower speed and `--gpuwrite-deflate-algo 0` for more speed, which continues to be the default.
- Fix edge case with FASTQ filtering.

Improvements spanning multiple tools

- Added the `--read-from-tmp-dir` option to germline pipeline and deepvariant_pipeline. It will run the variant caller in parallel with BAM generation. It has been tested on A100 and H100 but might cause out-of-memory on other GPUs.
- Added the `--fq2bamfast` option to germline pipeline, deepvariant_pipeline, and somatic pipeline to use fq2bamfast for alignment.

For further information see the Parabricks datasheet.

4.2.0-1 Release Notes

Several packages have been updated to GATK 4.3. Support for read lengths of up to 500 base pairs has been added to haplotypecaller and mutectcaller.

Several performance improvements.

A number of bugs have been fixed.

New Tools

Added markdup, a tool to locate and tag duplicate reads in a BAM or SAM file, where duplicate reads are defined as originating from a single fragment of DNA.

Improvements

deepvariant:

- Major performance improvements.

collectmultiplemetrics:

- Supports read lengths of up to 500 base pairs.

haplotypecaller:

- Updated to GATK 4.3.
- Supports read lengths of up to 500 base pairs.

- Adds support for the `--htvc-bam-output` option.
- `--batch` is deprecated.

mutectcaller:

- Updated to GATK 4.3.
- Supports read lengths of up to 500 base pairs.
- Adds support for the `--run-partition` option.

bammetrics

- Updated to GATK 4.3.

bqsr

- Clarified error messages.

Improvements spanning multiple tools

- Added more error checking when writing BAM files using `--gpuwrite`.
- Improved performance for BWA alignment, particularly for reads > 250 bases in length.
- Added GPUDirect Storage (GDS) support for fq2bam (BWA-MEM + GATK) and associated pipelines.
- Added a low memory mode for `--gpuwrite` (affects sorting).

Bug Fixes

- deepvariant Could only handle 32 alt alleles per candidate; can now handle up to 64.
- bamsort Bamsort could fail for certain coordinate sorts.

For further information see the Parabricks datasheet.

4.1.1-1 Release Notes

Bug Fixes

- Fixed a critical bug that makes germline pipeline run two times.
- Fixed --no-markdup option in rna_fq2bam
- Fixed a bug in rna_fq2bam when some values in bam are not aligned properly

For further information see the Parabricks datasheet.

4.1.0-1 Release Notes

Clara Parabricks 4.1.0-1 now supports GPUs with Ada Lovelace and Hopper architecture. The DeepVariant tool no longer needs different models for different GPUs. Please visit the updated Installation Requirements section for exact requirements.

We've also sped up several tools and are now using the GPU to sort and write BAM files in fq2bam, somatic, germline and deepvariant_germline.

Improvements

- Ada Lovelace and Hopper architecture GPUs are supported for all tools.
- fq2bam has been accelerated over the previous version by a significant factor.
- deepvariant has been upgraded to 1.5 and supports new PacBio options.
- deepvariant has been accelerated by a significant factor.
- Added cloud usage guides for:
    - AWS
    - DNANexus
    - GCP
    - Terra Bio
- Added information on how to get the best performance out of your Clara Parabricks software

Bug Fixes

- Fixed potential resource leaks in a few tools.

For further information see the Parabricks datasheet.

4.0.0-1 Release Notes

Clara Parabricks 4.0.0-1 is a major release with many significant changes. It streamlines how users get and set up the software and simplifies deployment on different platforms. If you are an existing Parabricks user, review this section to understand the major changes.

Licensing Changes

There is no license required to use NVIDIA Parabricks. The container works out of the box once downloaded.

Users who would like Enterprise Support can purchase NVIDIA AI Enterprise licenses, which provides full-stack support for Parabricks and many other NVIDIA software offerings.

To inquire about Enterprise Support for Parabricks, please reach out to the NVIDIA genomics team at https://www.nvidia.com/en-gb/clara/genomics/ ￼

Supported Tools

Starting with v4.0.0-1 the Clara Parabricks toolset focused primarily on those tools that benefit most from GPU acceleration.

If you would like access to one or more of the tools that are no longer available, please contact us in the developer forum.

WDL/Nextflow Workflows

Clara Parabricks containers are compatible with WDL and NextFlow for building customized workflows, intertwining GPU- and CPU-powered tasks with different compute requirements, and deploying at scale.

These enable workflows to be deployed on cloud batch services as well as local clusters (e.g. SLURM) in a well managed process, pulling from a combination of Parabricks and third-party containers and running these on pre-defined nodes.

../_images/Workflow%20diagram.png

For further information on running these workflows, and to see the open-source reference workflows, which can be easily forked/edited, visit the Clara Parabricks Workflows repository. This repository includes recommended instance configurations for deploying the GPU-based tools on cloud and can be easily forked/edited for your own purposes.

Improvements

- deepvariant now implements DeepVariant v1.5.
- haplotypecaller supports additional original HaplotypeCaller options.
- deepvariant and deepvariant_germline support the `--channel-insert-size` option.
- starfusion now adds `PG:Z:MarkDuplicates` to each output BAM record.

Bug Fixes

- Corrected rna_fq2bam sample code. The `--read-files-command zcat`, while not a required parameter, is needed for correct operation with compressed FASTQ files.
- Updated the list of supported haplotypecaller options.
- Corrected GATK sample code.
- Fixed an fq2bam bug that occurred when many Ns are present in the fastq files.
- General deepvariant performance improvements.

For further information see the Parabricks datasheet.

Getting Started with NVIDIA Parabricks

Installation Requirements

Hardware Requirements

- Any NVIDIA GPU that supports CUDA architecture 70, 75, 80, 86, 89 or 90 and has at least 16GB of GPU RAM. NVIDIA Parabricks has been tested on the following NVIDIA GPUs:
  - V100
  - T4
  - A10, A30, A40, A100, A6000
  - L4, L40
  - H100, H200
  - Grace Hopper Superchip
- The fq2bam tool requires at least 24 GB of GPU memory by default; the `--low-memory` option will reduce this to 16 GB of GPU memory at the cost of slower processing. All other tools require at least 16 GB of GPU memory per GPU.
- System Requirements:
  - A 2 GPU system should have at least 100GB CPU RAM and at least 24 CPU threads.
  - A 4 GPU system should have at least 196GB CPU RAM and at least 32 CPU threads.

- A 8 GPU system should have at least 392GB CPU RAM and at least 48 CPU threads.

ⓘ **Note**

Parabricks is supported on time-sliced virtual (vGPU) but not on Multi-Instance (MIG) GPUs.

Software Requirements

The following are software requirements for running Parabricks.

- An NVIDIA driver with version 525.60.13 or greater .
- Any Linux Operating System that supports nvidia-docker2 Docker version 20.10 (or higher)

Please see this page for more information on supported driver configurations.

ⓘ **Note**

Parabricks is available as Docker image. For Singularity users, please check here about importing a Docker image into a Singularity Image,

Verifying Hardware and Software Requirements

Checking available NVIDIA hardware and driver

To check your NVIDIA hardware and driver version, use the `nvidia-smi` command:

```
$ nvidia-smi +-----------------------------------------------------------------------------+ | NVIDIA-SMI
525.60.13 Driver Version: 525.60.13 CUDA Version: 12.0 | |-------------------------------+---
-------------------+----------------------+ | GPU Name Persistence-M| Bus-Id Disp.A | Volatile
Uncorr. ECC | | Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute
M. | | | | MIG M. |
|===============================+======================+=============
| 0 Tesla V100-DGXS... On | 00000000:07:00.0 Off | 0 | | N/A 44C P0 38W / 300W |
74MiB / 16155MiB | 0% Default | | | | N/A | +---------------------------------------------------
----------------------+ | Processes: | | GPU GI CI PID Type Process name GPU Memory |
| ID ID Usage |
|=============================================================================
| 0 N/A N/A 3019 G /usr/lib/xorg/Xorg 56MiB | +-------------------------------------------------
-------------------------+
```

This shows the following important information:

- The NVIDIA driver version is 525.60.13.
- The supported CUDA driver API is 12.0.
- The GPU has 16 GB of memory.

Checking available CPU RAM and threads

To see how much RAM and CPU threads in your machine, you can run the following:

```
# To check available memory $ cat /proc/meminfo | grep MemTotal # To check
available number of threads $ cat /proc/cpuinfo | grep processor | wc -l
```

Checking nvidia-docker2 installation

To make sure you have nvidia-docker2 installed, run this command:

```
$ docker run --rm --gpus all nvidia/cuda:12.0.0-base-ubuntu20.04 nvidia-smi
```

When it finishes downloading the container, it will run the `nvidia-smi` command and show you the same output as above.

Checking python version

To see which version of Python you have, enter the following command:

```
$ python3 --version
```

Make sure it's at least version 3 (3.6.9, 3.7, etc).

Getting the Software

The NVIDIA Parabricks Docker image can be obtained by running the following command:

```
$ docker pull nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1
```

At this point the software is ready to use.

Running NVIDIA Parabricks

From the Command Line

Parabricks is deployed using a Docker image. There are two parts to customizing a Parabricks run:

- Customizing Docker container specific options: These are the options that are passed to the `docker` command before the name of the container. For example, the user should mount their data directories within the Docker container by passing the `-v` option to Docker. See the Tutorials for more detailed examples.

- Parabricks specific options: These options are passed to the Parabricks command line to customize the Parabricks run. For example, you can choose which tool to run and pass tool-specific options.

For example, use the following command to run the Parabricks fq2bam (BWA-MEM + GATK) tool using a Docker container. See the tutorial for further details on how this command works.

```
$ docker run \ --gpus all \ --rm \ --volume $(pwd):/workdir \ --volume
$(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bam \ --
ref /workdir/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta \ --in-fq
/workdir/parabricks_sample/Data/sample_1.fq.gz
/workdir/parabricks_sample/Data/sample_2.fq.gz \ --out-bam
/outputdir/fq2bam_output.bam
```

Sample data is freely available. See the Getting The Sample Data section in the Tutorials for instructions on obtaining the sample data, and a step-by-step guide to using both fq2bam and Haplotype Caller.

Some useful Docker options to consider:

- `--gpus all` lets the Docker container use all the GPUs on the system. The GPUs available to Parabricks container can be limited using the `--gpus "device=&lt;list of GPUs&gt;"` option. Use `nvidia-smi` to see how many GPUs you have, and which one is which.

- `--rm` tells Docker to terminate the image once the command has finished.

- `--volume $(pwd):/image/data` mounts your current directory (a path on the server) on the Docker container in the `/image/data` directory (a path inside the Docker container). If your data is not in the current directory use an option similar to `--volume /path/to/your/data:/image/data`.

- `--workdir` tells Docker what working directory to execute the commands from (inside the container).

- The rest of the command is the Parabricks tool you want to run, followed by its arguments. For those familiar with pre-v4.0 versions of Parabricks and its `pbrun` command, this Docker invocation takes the place of `pbrun`.

Running Parabricks Using the Base Command Platform

An example command to launch a BaseCommand container on a single-GPU instance is:

```
ngc batch run --name "parabricks-germline" \ --instance dgxa100.80g.1.norm \ --
commandline "pbrun germline \ --ref
/workspace/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta \ --in-fq
/Data/HG002-NA24385-pFDA_S2_L002_R1_001-30x.fastq.gz /Data/HG002-NA24385-
pFDA_S2_L002_R2_001-30x.fastq.gz \ --knownSites
/workspace/parabricks_sample/Ref/Homo_sapiens_assembly38.known_indels.vcf.gz
\ --out-bam output.bam \ --out-variants output.vcf \ --out-recal-file report.txt \ --run-
partition \ --no-alt-contigs" \ --result /results \ --image "nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1"
```

Note that for other Parabricks commands (i.e. fq2bam, HaplotypeCaller, DeepVariant) the `ngc batch run` command is similar. Make sure to use the correct paths for your workplace or dataset that contains the data you intend to use.

Uninstalling the software

Uninstalling NVIDIA Parabricks is as simple as removing the Docker image.

```
$ docker images REPOSITORY TAG IMAGE ID CREATED SIZE ...
nvcr.io/nvidia/clara/clara-parabricks 4.3.1-1 516740210042 2 months ago 3.23GB ...
$ docker rmi 516740210042
```

The exact value of the "IMAGE ID" will vary depending on your installation.

ⓘ **Note**

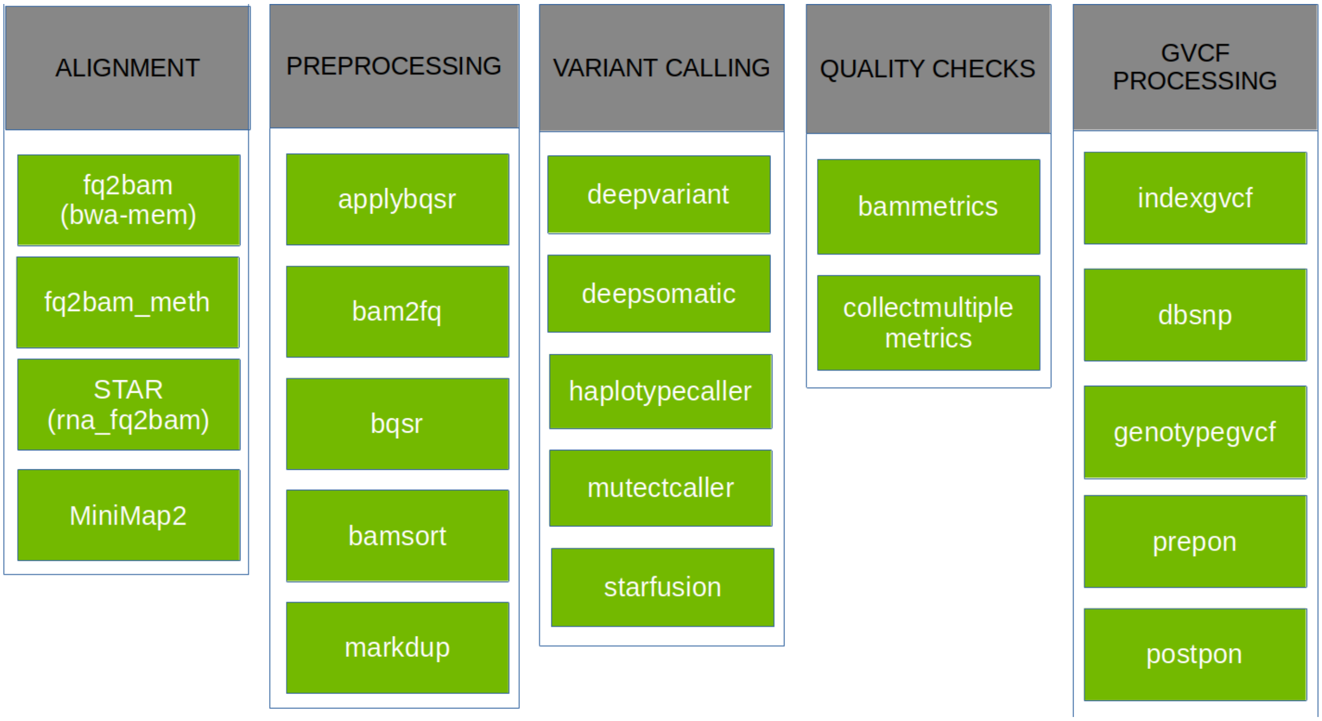
- User guides and Reference manuals can be found on the NVIDIA Parabricks documentation page.
- Answers to many other FAQs can be found on the developer forum.

Software Overview

NVIDIA Parabricks is a software suite for genomic analysis. It delivers major improvements in throughput time for common analytical tasks in genomics, including germline and somatic analysis. The core of the Parabricks software is its tight integration with the GPU, which takes raw data and transforms it according to the users requirements.

Parabricks has been tested on Dell, HPE, IBM, and NVIDIA servers at Amazon Web Services, Google Cloud, Oracle Cloud Infrastructure, and Microsoft Azure.

Parabricks supports the tools shown below:



ALIGNMENT	PREPROCESSING	VARIANT CALLING	QUALITY CHECKS	GVCF PROCESSING
fq2bam (bwa-mem)	applybqsr	deepvariant	bammetrics	indexgvcf
fq2bam_meth	bam2fq	deepsomatic	collectmultiple metrics	dbsnp
STAR (rna_fq2bam)	bqsr	haplotypecaller		genotypegvcf
MiniMap2	bamsort	mutectcaller		prepon
	markdup	starfusion		postpon

The somatic (Somatic Variant Caller), germline (GATK Germline Pipeline), pacbio_germline (Beta), and deepvariant_germline tools are actually a collection of several individual tools that are frequently run together, each grouped as a single command for the users convenience. For example, deepvariant_germline takes FASTA and FASTQ files as input and produces a VCF and BAM file as output. Internally, it runs BWA mem alignment, performs coordinate sorting, marks duplicates, and then runs DeepVariant.

The Tool Reference page lists all the individual tools. The Parabricks WDL/Nextflow Workflows page discusses the use of Parabricks with WDL and Nextflow. The Compatible CPU Software Versions page lists the open-source CPU tools Parabricks is compatible with.

- Software Tools
- NVIDIA Parabricks WDL/Nextflow Workflows
- Compatible CPU Software Versions

Software Tools

The following tools are available in the NVIDIA Parabricks software. Click on a tool name for tool-specific options.

The Parabricks somatic (Somatic Variant Caller), germline (GATK Germline Pipeline) and deepvariant_germline tools are collections of several other individual tools that are commonly used together, all wrapped up as a single tool. For example, the deepvariant_germline takes FASTA and FASTQ files as input and produces a VCF and BAM file as output. Internally, it runs BWA mem alignment, performs coordinate sorting, marks duplicates, and then runs DeepVariant.

Tool	Details
applybqsr	Apply BQSR report to a BAM file and generate a new BAM file
bam2fq	Convert a BAM file to FASTQ
bammetrics	Collect WGS Metrics on a BAM file
bamsort	Sort a BAM file
bqsr	Collect BQSR report on a BAM file
collectmultiplemetrics	Collect multiple classes of metrics on a BAM file
dbsnp	Annotate variants based on a dbsnp
deepsomatic	Run GPU-DeepSomatic for calling somatic variants
deepvariant	Run GPU-DeepVariant for calling germline variants
deepvariant_germline	Run the germline pipeline from FASTQ to VCF using a deep neural network analysis
fq2bam (BWA-MEM + GATK)	Run bwa mem, co-ordinate sorting, marking duplicates, and Base Quality Score Recalibration
fq2bam_meth	Run GPU-accelerated bwa-meth compatible alignment, co-ordinate sorting, marking duplicates, and Base Quality Score Recalibration

fq2bamfast (BWA-MEM + GATK)	Run newly optimized version of bwa mem, co-ordinate sorting, marking duplicates, and Base Quality Score Recalibration
genotypegvcf	Convert a GVCF to VCF
germline (GATK Germline Pipeline)	Run the germline pipeline from FASTQ to VCF
haplotypecaller	Run GPU-HaplotypeCaller for calling germline variants
indexgvcf	Index a GVCF file
markdup	Identifies duplicate reads
minimap2 (Beta)	Align long read sequences against a large reference database to convert FASTQ to BAM/CRAM
mutectcaller	Run GPU-Mutect2 for tumor-normal analysis
pacbio_germline (Beta)	Run the germline pipeline from FASTQ to VCF by aligning long read sequences with minimap2 and using a deep neural network analysis
postpon	Generate the final VCF output of doing mutect pon
prepon	Build an index for PON file, which is the prerequisite to performing mutect pon
rna_fq2bam	Run RNA-seq data through the fq2bam pipeline
somatic (Somatic Variant Caller)	Run the somatic pipeline from FASTQ to VCF
starfusion	Identify candidate fusion transcripts supported by Illumina reads

NVIDIA Parabricks WDL/Nextflow Workflows

Parabricks containers are compatible with WDL and NextFlow for building customized workflows, intertwining GPU- and CPU-powered tasks with different compute requirements, and deploying at scale.

These enable workflows to be deployed on cloud batch services as well as local clusters (e.g. SLURM) in a well managed process, pulling from a combination of Parabricks and third-party containers and running these on pre-defined nodes.

../../_images/Workflow%20diagram.png

For further information on running these workflows, and to see the open-source reference workflows, which can be easily forked/edited, visit the Parabricks Workflows repository. This repository includes recommended instance configurations for deploying the GPU-based tools on cloud and can be easily forked/edited for your own purposes.

Compatible CPU Software Versions

Parabricks produces the same results as the following tools:

Tool	Version
BWA	0.7.15
bwa-meth	0.2.7
Deepvariant	1.6.1
GATK	4.3.0.0
minimap2	2.26
pbmm2	1.13.0
STAR	2.7.2a
STAR-Fusion	1.7.0

Best Performance

NVIDIA Parabricks software can give very high performance when all the required computing resources are provided to it. It should meet all the requirements in Installation Requirements section. Here are a few examples of how to get Parabricks software to give its best performance.

See the Hardware Requirements section for minimum hardware requirements.

See the Software Requirements section for minimum software requirements.

Basic Performance Tuning

The goal of the NVIDA Parabricks software is to get the highest performance for bioinformatics and genomic analysis. There are a few key, basic system options that you can tune to achieve maximum performance.

Use a Fast SSD

Parabricks software operates with two kinds of files:

- Input/output files specified by the user
- Temporary files created during execution and deleted at the end of the run

The best performance is achieved when both kinds of files are on a fast, local SSD. If this is not possible you can place the input/output files on a fast network storage device and the temporary files on a local SSD using the `--tmp-dir` option.

> **Note**
>
> Tests have shown that you can use up to 4 GPUs and still get good performance with the Lustre network for Input/Output files. If you plan to use more than 4 GPUs, we highly recommend using local SSDs for all kinds of files.

DGX Users

The DGX comes with a SSD, usually mounted on `/raid`. Use this disk, and use a directory on this disk as the `--tmp-dir`. For initial testing, you can even copy the input files to this disk to eliminate variability in performance.

In certain cases, Transparent HugePage Support (THP) has been found to increase performance. Consider enabling THP and testing performance on benchmark cases.

Specifying which GPUs to use

You can choose the number of GPUs to run using the command line option `--num-gpus N` for those tools that use GPUs. With this option only the first `N` GPUs listed in the output of `nvidia-smi` will be used.

To use specific GPUs set the environment variable `NVIDIA_VISIBLE_DEVICES`. GPUs are numbered starting with zero. For example, this command will use only the second (GPU #1) and fourth (GPU #3) GPUs:

```
$ NVIDIA_VISIBLE_DEVICES="1,3" pbrun fq2bam --num-gpus 2 --ref Ref.fa --in-fq
S1_1.fastq.gz --in-fq S1_2.fastq.gz
```

Tool-Specific Performance Guidelines

This section details guidelines specific to individual tools.

Best Performance for Germline Pipeline

On an H100 DGX the gremline pipeline typically runs in under ten minutes.

For backwards-compatible results with less performance when using `--fq2bamfast`, set `--bwa-options="-K 10000000"`.

```
$ # 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 \ --workdir /workdir \ --env
TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456 \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun germline \ --ref
/workdir/Homo_sapiens_assembly38.fasta \ --in-fq /workdir/fastq1.gz
/workdir/fastq2.gz \ --out-bam /outputdir/fq2bam_output.bam \ --tmp-dir /workdir \
--num-cpu-threads-per-stage 16 \ --bwa-cpu-thread-pool 16 \ --out-variants
/outputdir/out.vcf \ --run-partition \ --read-from-tmp-dir \ --gpusort \ --gpuwrite \ --
fq2bamfast \ --keep-tmp
```

Best Performance for Deepvariant Germline Pipeline

For backwards-compatible results with less performance when using `--fq2bamfast`, set `--bwa-options="-K 10000000"`.

```
$ # 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 \ --workdir /workdir \ --env
TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456 \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun deepvariant_germline \ --ref
/workdir/Homo_sapiens_assembly38.fasta \ --in-fq /workdir/fastq1.gz
/workdir/fastq2.gz \ --out-bam /outputdir/fq2bam_output.bam \ --tmp-dir /workdir \
--num-cpu-threads-per-stage 16 \ --bwa-cpu-thread-pool 16 \ --out-variants
/outputdir/out.vcf \ --run-partition \ --read-from-tmp-dir \ --num-streams-per-gpu 4 \
--gpusort \ --gpuwrite \ --fq2bamfast \ --keep-tmp
```

Best Performance for PacBio Germline Pipeline

```
$ # 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 \ --workdir /workdir --env TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456 nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun pacbio_germline \ --ref /workdir/${REFERENCE_FILE} \ --in-fq /workdir/${INPUT_FASTQ} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-variants /outputdir/out.vcf --num-chaining-threads 3 \ --alignment-large-pair-size 5000 \ --process-large-alignments-on-cpu \ --num-alignment-threads-per-gpu 8 \ --num-alignment-device-mem-buffers 8 \ --run-partition \ --read-from-tmp-dir \ --num-streams-per-gpu 4 \ --gpusort \ --gpuwrite \ --keep-tmp
```

Best Performance for fq2bam/fq2bamfast

Use the new beta version, fq2bamfast. For backwards-compatible results with less performance, set `--bwa-options="-K 10000000"`.

```
$ # 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 \ --workdir /workdir --env TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456 \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bamfast \ --ref /workdir/Homo_sapiens_assembly38.fasta \ --in-fq /workdir/fastq1.gz /workdir/fastq2.gz \ --out-bam /outputdir/fq2bam_output.bam \ --tmp-dir /workdir \ --bwa-cpu-thread-pool 16 \ --out-recal-file recal.txt \ --knownSites /workdir/hg.known_indels.vcf \ --gpusort \ --gpuwrite
```

Best Performance for deepvariant

DeepVariant from Parabricks has the ability to use multiple streams on a GPU. The number of streams that can be used depends on the available resources. The default number of streams is set to two but can be increased up to a maximum of six to get better performance. This is something that has to be experimented with, before getting the optimal number on your system.

```
$ # 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 \ --workdir /workdir --env
TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun deepvariant \ --ref
/workdir/Homo_sapiens_assembly38.fasta \ --in-bam
/outputdir/fq2bam_output.bam \ --out-variants /outputdir/out.vcf \ --num-streams-per-gpu
4 \ --run-partition
```

Best Performance for haplotypecaller

```
$ # 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 \ --workdir /workdir --env
TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun haplotypecaller \ --ref
/workdir/Homo_sapiens_assembly38.fasta \ --in-bam
/outputdir/fq2bam_output.bam \ --out-variants /outputdir/out.vcf \ --num-htvc-threads
8 \ --no-alt-contigs \#This flag will ignore all outputs after chrM --run-partition
```

Best Performance for minimap2

The following command line options provided optimal performance for PacBio data running on 2x 7742 + 2x A100 80GB PCIe.

```
$ # 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 \ --workdir /workdir --env
TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun minimap2 \ --ref
/workdir/${REFERENCE_FILE} \ --in-fq /workdir/${INPUT_FASTQ} \ --out-bam
/outputdir/${OUTPUT_BAM} \ --num-chaining-threads 3 \ --alignment-large-pair-size
5000 \ --process-large-alignments-on-cpu \ --num-alignment-threads-per-gpu 8 \
--num-alignment-device-mem-buffers 8 \ --gpusort \ --gpuwrite
```

GDS Support

For additional performance improvements and final BAM writing bandwidth use GPUDirect Storage (GDS), part of the CUDA toolkit. Note that the system must be set up and supported to use GDS.

The following are references for setting up and using GDS:

- Overall NVIDIA GPUDirect Storage documentation
- NVIDIA GPUDirect Storage Installation and Troubleshooting Guide
- Using GDS in containers

```
# Using GDS with the convenience docker wrapper. $ wget
https://raw.githubusercontent.com/NVIDIA/MagnumIO/main/gds/docker/gds-run-container $ chmod +x gds-run-container $ ./gds-run-container run \ --rm \ --gpus all
\ --enable-mofed \ --enable-gds \ --volume INPUT_DIR:/workdir \ --volume
OUTPUT_DIR:/outputdir \ --workdir /workdir \ --env
TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456 \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bam \ --ref
/workdir/Homo_sapiens_assembly38.fasta \ --in-fq /workdir/fastq1.gz
/workdir/fastq2.gz \ --out-bam /outputdir/fq2bam_output.bam \ --tmp-dir /workdir \
--out-recal-file recal.txt \ --knownSites /workdir/hg.known_indels.vcf \ --gpusort \
--gpuwrite \ --use-gds # Using GDS without the wrapper. $ docker run \ --ipc host \
--volume /run/udev:/run/udev:ro \ --device=/dev/nvidia-fs0 \ --device=/dev/nvidia-fs1 \
--device=/dev/nvidia-fs2 \ --device=/dev/nvidia-fs3 \ --device=/dev/nvidia-fs4 \
--device=/dev/nvidia-fs5 \ --device=/dev/nvidia-fs6 \ --device=/dev/nvidia-fs7 \
--device=/dev/nvidia-fs8 \ --device=/dev/nvidia-fs9 \ --device=/dev/nvidia-fs10 \
--device=/dev/nvidia-fs11 \ --device=/dev/nvidia-fs12 \ --device=/dev/nvidia-fs13 \
--device=/dev/nvidia-fs14 \ --device=/dev/nvidia-fs15 \ --rm \ --gpus all \ -enable-mofed \ --volume INPUT_DIR:/workdir \ --volume OUTPUT_DIR:/outputdir \ --workdir
/workdir \ --env TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES=268435456 \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bam --ref
/workdir/Homo_sapiens_assembly38.fasta \ --in-fq /workdir/fastq1.gz
/workdir/fastq2.gz \ --out-bam /outputdir/fq2bam_output.bam \ --tmp-dir /workdir \
--out-recal-file recal.txt \ --knownSites /workdir/hg.known_indels.vcf \ --gpusort \ --
gpuwrite \ --use-gds
```

Tutorials

The tutorials walk you through a simple use case for NVIDIA Parabricks, giving a brief introduction of how it works. You will start by downloading some sample data:

- A reference file (`Homo_sapiens_assembly38.fasta`) and its index
- A 'known indels' file and its index
- Two FASTQ files
- Associated index files
- One bam file for markdup

The tutorials then walk through the following steps:

- Alignment (FASTA + FASTQ ==> BAM)
- Variant calling (BAM ==> VCF)

The tutorials are meant to be simple and straightforward and to only cover a single, specific use case. You should be able to copy and paste the commands into a terminal window and get the same results as shown. The How-Tos cover more general problem solving using Parabricks.

Steps in the Tutorial

- Getting The Sample Data
- FQ2BAM Tutorial
- HaplotypeCaller Tutorial
- Cloud Usage Guides
- DeepVariant training using Parabricks

Getting The Sample Data

Download the sample data with the following command. This data will be used in the tutorial examples.

```
$ wget -O parabricks_sample.tar.gz \
"https://s3.amazonaws.com/parabricks.sample/parabricks_sample.tar.gz"
```

Note

The tar file is 10.6GB and, when extracted, an additional 15GB.

Extract the data with this command:

```
$ tar xvf parabricks_sample.tar.gz parabricks_sample/ parabricks_sample/Data/
parabricks_sample/Data/markdup_input.bam
parabricks_sample/Data/sample_2.fq.gz parabricks_sample/Data/sample_1.fq.gz
parabricks_sample/Ref/ parabricks_sample/Ref/Homo_sapiens_assembly38.fasta
parabricks_sample/Ref/Homo_sapiens_assembly38.fasta.pac
parabricks_sample/Ref/Homo_sapiens_assembly38.fasta.ann
parabricks_sample/Ref/Homo_sapiens_assembly38.known_indels.vcf.gz.tbi
parabricks_sample/Ref/Homo_sapiens_assembly38.fasta.amb
parabricks_sample/Ref/Homo_sapiens_assembly38.dict
parabricks_sample/Ref/Homo_sapiens_assembly38.fasta.fai
parabricks_sample/Ref/Homo_sapiens_assembly38.known_indels.vcf.gz
parabricks_sample/Ref/Homo_sapiens_assembly38.fasta.bwt
parabricks_sample/Ref/Homo_sapiens_assembly38.fasta.sa
```

You are now ready to try the example commands shown in the rest of the Tutorials section.

FQ2BAM Tutorial

This tutorial will show you how to run our core alignment tool, FQ2BAM, which allows you to align a FASTQ file according to GATK best practices at blazing speeds. This includes the gold-standard alignment tool BWA-MEM with inbuilt co-ordinate sorting of the output file, and optionally application of base-quality-score-recalibration and marking of duplicate reads.

The `fq2bam` tool aligns, sorts (by coordinate), and marks duplicates in paired-end FASTQ file data. The data files used in this example are taken from the sample data downloaded in the previous section.

If you execute the following command using the NVIDIA Parabricks sample data, you should get the same results as shown here.

Before executing this command, make sure your current directory is where you extracted the sample data; it should have a **parabricks_sample** sub-directory.

```
$ docker run \ --gpus all \ --rm \ --volume $(pwd):/workdir \ --volume
$(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bam \ --
ref /workdir/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta \ --in-fq
/workdir/parabricks_sample/Data/sample_1.fq.gz
/workdir/parabricks_sample/Data/sample_2.fq.gz \ --out-bam
/outputdir/fq2bam_output.bam [Parabricks Options Mesg]: Checking argument
compatibility [Parabricks Options Mesg]: Automatically generating ID prefix
[Parabricks Options Mesg]: Read group created for
/workdir/parabricks_sample/Data/sample_1.fq.gz and
/workdir/parabricks_sample/Data/sample_2.fq.gz [Parabricks Options Mesg]:
@RG\tID:HK3TJBCX2.1\tLB:lib1\tPL:bar\tSM:sample\tPU:HK3TJBCX2.1 [PB Info 2022-
Sep-02 19:49:27] -------------------------------------------------------------------------------- [PB Info
2022-Sep-02 19:49:27] || Parabricks accelerated Genomics Pipeline || [PB Info
2022-Sep-02 19:49:27] || Version 4.0.0-1 || [PB Info 2022-Sep-02 19:49:27] || GPU-
BWA mem, Sorting Phase-I || [PB Info 2022-Sep-02 19:49:27] --------------------------------
------------------------------------------------ [M::bwa_idx_load_from_disk] read 0 ALT contigs [PB
Warning 2022-Sep-02 19:50:02][ParaBricks/src/pbOpts.cu:325] WARNING The
system has 12 threads, however recommended number of threads with 1 GPU is
```

```
16. The run might not finish or might have less than expected performance. [PB Info
2022-Sep-02 19:50:02] GPU-BWA mem [PB Info 2022-Sep-02 19:50:02]
ProgressMeter Reads Base Pairs Aligned [PB Info 2022-Sep-02 19:50:45] 5043564
580000000 [PB Info 2022-Sep-02 19:51:21] 10087128 1160000000 [PB Info 2022-
Sep-02 19:51:59] 15130692 1740000000 [PB Info 2022-Sep-02 19:52:39] 20174256
2320000000 [PB Info 2022-Sep-02 19:53:20] 25217820 2900000000 [PB Info 2022-
Sep-02 19:53:58] 30261384 3480000000 [PB Info 2022-Sep-02 19:54:36] 35304948
4060000000 [PB Info 2022-Sep-02 19:55:13] 40348512 4640000000 [PB Info 2022-
Sep-02 19:55:53] 45392076 5220000000 [PB Info 2022-Sep-02 19:56:36] 50435640
5800000000 [PB Info 2022-Sep-02 19:57:02] GPU-BWA Mem time: 420.426442
seconds [PB Info 2022-Sep-02 19:57:02] GPU-BWA Mem is finished. [main] CMD:
/usr/local/parabricks/binaries//bin/bwa mem -Z ./pbOpts.txt
/workdir/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta
/workdir/parabricks_sample/Data/sample_1.fq.gz
/workdir/parabricks_sample/Data/sample_2.fq.gz
@RG\tID:HK3TJBCX2.1\tLB:lib1\tPL:bar\tSM:sample\tPU:HK3TJBCX2.1 [main] Real
time: 455.468 sec; CPU: 4766.384 sec [PB Info 2022-Sep-02 19:57:02] ---------------------
------------------------------------------------------ [PB Info 2022-Sep-02 19:57:02] || Program:
GPU-BWA mem, Sorting Phase-I || [PB Info 2022-Sep-02 19:57:02] || Version: 4.0.0-
1 || [PB Info 2022-Sep-02 19:57:02] || Start Time: Fri Sep 2 19:49:27 2022 || [PB
Info 2022-Sep-02 19:57:02] || End Time: Fri Sep 2 19:57:02 2022 || [PB Info 2022-
Sep-02 19:57:02] || Total Time: 7 minutes 35 seconds || [PB Info 2022-Sep-02
19:57:02] ------------------------------------------------------------------------ [PB Info 2022-Sep-
02 19:57:03] ------------------------------------------------------------------------ [PB Info 2022-
Sep-02 19:57:03] || Parabricks accelerated Genomics Pipeline || [PB Info 2022-Sep-
02 19:57:03] || Version 4.0.0-1 || [PB Info 2022-Sep-02 19:57:03] || Sorting Phase-II
|| [PB Info 2022-Sep-02 19:57:03] ------------------------------------------------------------------
------ [PB Info 2022-Sep-02 19:57:03] progressMeter - Percentage [PB Info 2022-Sep-
02 19:57:03] 0.0 0.00 GB [PB Info 2022-Sep-02 19:57:13] 72.8 0.00 GB [PB Info 2022-
Sep-02 19:57:23] Sorting and Marking: 20.001 seconds [PB Info 2022-Sep-02
19:57:23] ------------------------------------------------------------------------ [PB Info 2022-Sep-
02 19:57:23] || Program: Sorting Phase-II || [PB Info 2022-Sep-02 19:57:23] ||
Version: 4.0.0-1 || [PB Info 2022-Sep-02 19:57:23] || Start Time: Fri Sep 2 19:57:03
2022 || [PB Info 2022-Sep-02 19:57:23] || End Time: Fri Sep 2 19:57:23 2022 || [PB
Info 2022-Sep-02 19:57:23] || Total Time: 20 seconds || [PB Info 2022-Sep-02
19:57:23] ------------------------------------------------------------------------ [PB Info 2022-Sep-
02 19:57:23] ------------------------------------------------------------------------ [PB Info 2022-
Sep-02 19:57:23] || Parabricks accelerated Genomics Pipeline || [PB Info 2022-Sep-
02 19:57:23] || Version 4.0.0-1 || [PB Info 2022-Sep-02 19:57:23] || Marking
Duplicates, BQSR || [PB Info 2022-Sep-02 19:57:23] ------------------------------------------
------------------------------ [PB Info 2022-Sep-02 19:57:24] progressMeter - Percentage
[PB Info 2022-Sep-02 19:57:34] 13.6 16.60 GB [PB Info 2022-Sep-02 19:57:44] 31.1
13.45 GB [PB Info 2022-Sep-02 19:57:54] 46.8 10.22 GB [PB Info 2022-Sep-02
19:58:04] 61.1 7.05 GB [PB Info 2022-Sep-02 19:58:14] 77.3 3.84 GB [PB Info 2022-
Sep-02 19:58:24] 91.4 0.60 GB [PB Info 2022-Sep-02 19:58:34] 100.0 0.00 GB [PB Info
2022-Sep-02 19:59:18] BQSR and writing final BAM: 113.592 seconds [PB Info 2022-
Sep-02 19:59:18] ------------------------------------------------------------------------ [PB Info
2022-Sep-02 19:59:18] || Program: Marking Duplicates, BQSR || [PB Info 2022-Sep-
02 19:59:18] || Version: 4.0.0-1 || [PB Info 2022-Sep-02 19:59:18] || Start Time: Fri
Sep 2 19:57:23 2022 || [PB Info 2022-Sep-02 19:59:18] || End Time: Fri Sep 2
19:59:18 2022 || [PB Info 2022-Sep-02 19:59:18] || Total Time: 1 minute 55 seconds
|| [PB Info 2022-Sep-02 19:59:18] ------------------------------------------------------------------
------ Please visit https://docs.nvidia.com/clara/#parabricks for detailed
documentation
```

On an AWS g4dn.8xlarge instance (32 vCPUs, one T4 GPU, 128 GB memory), this takes approximately six minutes.

If you get an out-of-memory error make sure your computer has enough RAM, and that large amounts of memory aren't being used by other programs.

This `fq2bam` command produces three output files:

```
$ ls -l total 14330820 -rw-r--r-- 1 root root 4819386804 Sep 2 15:58
fq2bam_output.bam -rw-r--r-- 1 root root 6882792 Sep 2 15:59
fq2bam_output.bam.bai -rw-r--r-- 1 root root 87690 Sep 2 15:59
fq2bam_output_chrs.txt (input files not shown)
```

The first line of `fq2bam_output.bam` (as viewed with the `samtools view fq2bam_output.bam` command) is as follows:

```
HWI-D00127:570:HK3TJBCX2:1:1202:9643:76055 99 chr1 10027 26 24M5I86M =
10178 231
ACCCTAACCCTAACCCTAACCCGACCCCGACCCCGACCCAAACCCAAACCCTAACCCTAACCCT
DDDDDHGHIIIIIHIIHHIHHHHIHIIIIIIHDHHIHHHHIHIHIIIIFHIEHHIIHHIIIIEHIIIIHHIHIIICHE@1F
1GEFE1111D11<FH11<FD11<<FFE111<11 MD:Z:22T5T0A4T5T41A27
PG:Z:MarkDuplicatesRG:Z:HK3TJBCX2.1 NM:i:11 AS:i:69 XS:i:72 ....
```

Note

If the `fq2bam` command is run on a system with too little memory, you will see this message after the initial header:

WARNING The system has 62 GB, however recommended RAM with 1 GPU is 64 GB. The run might not finish or might have less than expected performance.

HaplotypeCaller Tutorial

This tutorial will show you how to run the gold-standard GATK variant caller, HaplotypeCaller, which takes your aligned output BAM from the FQ2BAM Tutorial, assembles plausible haplotypes from active regions, and identifies genotype likelihoods according to Bayes' Rule. The result is a Variant Call Format (VCF) file of all the variant calls across the genome, including their position in the genome and the allelic information.

To do this, run the following command:

```
$ docker run \ --gpus all \ --rm \ --volume $(pwd):/workdir \ --volume
$(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun
haplotypecaller \ --ref
/workdir/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta \ --in-bam
/workdir/fq2bam_output.bam \ --out-variants /outputdir/variants.vcf [PB Info 2022-
Sep-02 20:08:13] ------------------------------------------------------------------------------ [PB Info
2022-Sep-02 20:08:13] || Parabricks accelerated Genomics Pipeline || [PB Info
2022-Sep-02 20:08:13] || Version 4.0.0-1 || [PB Info 2022-Sep-02 20:08:13] || GPU-
GATK4 HaplotypeCaller || [PB Info 2022-Sep-02 20:08:13] ------------------------------------
---------------------------------------- [PB Info 2022-Sep-02 20:08:48] 0
/outputdir/fq2bam_output.bam/outputdir/variants.vcf [PB Info 2022-Sep-02
20:08:48] ProgressMeter - Current-Locus Elapsed-Minutes Regions-Processed
Regions/Minute [PB Info 2022-Sep-02 20:08:58] chr1:26179201 0.2 143431 860586
[PB Info 2022-Sep-02 20:09:08] chr1:88982401 0.3 428734 1286202 [PB Info 2022-
Sep-02 20:09:18] chr1:155342401 0.5 622031 1244062 [PB Info 2022-Sep-02
20:09:28] chr1:222508801 0.7 921183 1381774 [PB Info 2022-Sep-02 20:09:38]
chr2:38697601 0.8 1206544 1447852 [PB Info 2022-Sep-02 20:09:48]
chr2:101697601 1.0 1469587 1469585 [PB Info 2022-Sep-02 20:09:58]
chr2:176107201 1.2 1802198 1544741 [PB Info 2022-Sep-02 20:10:08] chr3:8990401
1.3 2124644 1593483 [PB Info 2022-Sep-02 20:10:18] chr3:55473601 1.5 2349628
1566418 [PB Info 2022-Sep-02 20:10:28] chr3:133680001 1.7 2671435 1602861 [PB
Info 2022-Sep-02 20:10:38] chr3:192724801 1.8 2922628 1594160 [PB Info 2022-
Sep-02 20:10:48] chr4:61934401 2.0 3222268 1611134 [PB Info 2022-Sep-02
20:10:58] chr4:143524801 2.2 3573491 1649303 [PB Info 2022-Sep-02 20:11:08]
chr5:38947201 2.3 3949926 1692825 [PB Info 2022-Sep-02 20:11:18]
chr5:116337601 2.5 4255653 1702261 [PB Info 2022-Sep-02 20:11:28]
chr5:176035201 2.7 4508018 1690506 [PB Info 2022-Sep-02 20:11:38]
chr6:54768001 2.8 4780747 1687322 [PB Info 2022-Sep-02 20:11:48]
chr6:152140801 3.0 5176363 1725454 [PB Info 2022-Sep-02 20:11:58]
chr7:27105601 3.2 5377117 1698036 [PB Info 2022-Sep-02 20:12:08]
chr7:105105601 3.3 5713863 1714158 [PB Info 2022-Sep-02 20:12:18]
chr8:18086401 3.5 6002114 1714889 [PB Info 2022-Sep-02 20:12:28] chr8:73915201
3.7 6267992 1709451 [PB Info 2022-Sep-02 20:12:38] chr9:5553601 3.8 6588617
1718769 [PB Info 2022-Sep-02 20:12:48] chr9:93672001 4.0 6886981 1721745 [PB
Info 2022-Sep-02 20:12:58] chr10:4094401 4.2 7095584 1702939 [PB Info 2022-Sep-
02 20:13:08] chr10:94593601 4.3 7489282 1728295 [PB Info 2022-Sep-02 20:13:18]
chr11:18398401 4.5 7757420 1723871 [PB Info 2022-Sep-02 20:13:28]
chr11:95976001 4.7 8083942 1732273 [PB Info 2022-Sep-02 20:13:38]
chr12:6652801 4.8 8282324 1713584 [PB Info 2022-Sep-02 20:13:48]
chr12:70632001 5.0 8560282 1712056 [PB Info 2022-Sep-02 20:13:58]
chr13:24446401 5.2 8860588 1714952 [PB Info 2022-Sep-02 20:14:08]
chr13:99038401 5.3 9200934 1725175 [PB Info 2022-Sep-02 20:14:18]
chr14:68467201 5.5 9480546 1723735 [PB Info 2022-Sep-02 20:14:28]
```

```
chr15:66432001 5.7 9821985 1733291 [PB Info 2022-Sep-02 20:14:38]
chr16:31128001 5.8 10123582 1735471 [PB Info 2022-Sep-02 20:14:48]
chr17:15782401 6.0 10402640 1733773 [PB Info 2022-Sep-02 20:14:58]
chr17:55262401 6.2 10553414 1711364 [PB Info 2022-Sep-02 20:15:08]
chr18:27960001 6.3 10790487 1703761 [PB Info 2022-Sep-02 20:15:18]
chr19:15883201 6.5 11074759 1703809 [PB Info 2022-Sep-02 20:15:28]
chr20:16108801 6.7 11311965 1696794 [PB Info 2022-Sep-02 20:15:38]
chr21:10468801 6.8 11563626 1692237 [PB Info 2022-Sep-02 20:15:48]
chr22:29289601 7.0 11766272 1680896 [PB Info 2022-Sep-02 20:15:58]
chrX:77452801 7.2 12197330 1701952 [PB Info 2022-Sep-02 20:16:08]
chrUn_JTFH01000876v1_decoy:1 7.3 12604293 1718767 [PB Info 2022-Sep-02
20:16:18] Total time taken: 450.258102 [PB Info 2022-Sep-02 20:16:18] -------------------
--------------------------------------------------------- [PB Info 2022-Sep-02 20:16:18] ||
Program: GPU-GATK4 HaplotypeCaller || [PB Info 2022-Sep-02 20:16:18] || Version:
4.0.0-1 || [PB Info 2022-Sep-02 20:16:18] || Start Time: Fri Sep 2 20:08:13 2022 ||
[PB Info 2022-Sep-02 20:16:18] || End Time: Fri Sep 2 20:16:18 2022 || [PB Info
2022-Sep-02 20:16:18] || Total Time: 8 minutes 5 seconds || [PB Info 2022-Sep-02
20:16:18] ---------------------------------------------------------------------------
/usr/local/parabricks/binaries//bin/htvc
/workdir/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta
/outputdir/fq2bam_output.bam 1 -o /outputdir/variants.vcf -nt 5 Please visit
https://docs.nvidia.com/clara/#parabricks for detailed documentation
```

If you get an out-of-memory error make sure your computer has enough RAM, and that large amounts of memory aren't being used by other programs.

You should now have the following files in your directory:

```
$ ls -lrt -rw-r--r-- 1 root root 4819386804 Sep 2 15:58 fq2bam_output.bam -rw-r--r-- 1
root root 6882792 Sep 2 15:59 fq2bam_output.bam.bai -rw-r--r-- 1 root root 87690
Sep 2 15:59 fq2bam_output_chrs.txt -rw-r--r-- 1 root root 23643404 Sep 2 16:16
variants.vcf (sample data not shown)
```

The first ten non-header lines of `variants.vcf` should be as follows:

```
chr1 16378 . T C 45.28 .
AC=2;AF=1.00;AN=2;DP=2;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=23
GT:AD:DP:GQ:PL 1/1:0,2:2:6:57,6,0 chr1 63268 . T C 43.28 .
AC=2;AF=1.00;AN=2;DP=2;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=23
GT:AD:DP:GQ:PL 1/1:0,2:2:6:55,6,0 chr1 63516 . A G 1202.03 .
AC=2;AF=1.00;AN=2;DP=40;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=26
GT:AD:DP:GQ:PL 1/1:0,40:40:99:1216,120,0 chr1 63527 . T C 1002.03 .
AC=2;AF=1.00;AN=2;DP=33;ExcessHet=3.0103;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=26
GT:AD:DP:GQ:PL 1/1:0,33:33:99:1016,99,0 chr1 131609 . C A 83.60 .
AC=1;AF=0.500;AN=2;BaseQRankSum=-0.275;DP=11;ExcessHet=3.0103;FS=0.000;MLEA
GT:AD:DP:GQ:PL 0/1:6,5:11:91:91,0,115 chr1 133483 . G T 70.60 .
AC=1;AF=0.500;AN=2;BaseQRankSum=-1.534;DP=8;ExcessHet=3.0103;FS=3.332;MLEAC
GT:AD:DP:GQ:PL 0/1:5,3:8:78:78,0,100 chr1 264627 . A G 37.28 .
AC=2;AF=1.00;AN=2;DP=2;ExcessHet=3.0103;FS=0.000;MLEAC=1;MLEAF=0.500;MQ=49
GT:AD:DP:GQ:PL 1/1:0,2:2:6:49,6,0 chr1 268012 . G A 330.60 .
AC=1;AF=0.500;AN=2;BaseQRankSum=0.636;DP=116;ExcessHet=3.0103;FS=1.639;MLEA
GT:AD:DP:GQ:PL 0/1:93,23:116:99:338,0,3307 chr1 268130 . G T 2188.60 .
AC=1;AF=0.500;AN=2;BaseQRankSum=1.543;DP=228;ExcessHet=3.0103;FS=8.632;MLEA
GT:AD:DP:GQ:PL 0/1:128,100:228:99:2196,0,4482 chr1 268516 . C T 59.60 .
AC=1;AF=0.500;AN=2;BaseQRankSum=1.465;DP=7;ExcessHet=3.0103;FS=3.680;MLEAC
GT:AD:DP:GQ:PL 0/1:5,2:7:67:67,0,138
```

Cloud Usage Guides

In this section we present several step-by-step guides to running NVIDIA Parabricks on several cloud platforms.

- Running NVIDIA Parabricks on AWS
  - What is NVIDIA Parabricks?
  - Starting an EC2 Instance
  - Installing Parabricks
  - Testing Parabricks
  - Private Workflows
  - Closing Remarks
- Running NVIDIA Parabricks on Azure

  - What is NVIDIA Parabricks?
  - Starting a Compute Instance
  - Installing Parabricks
  - Testing Parabricks
  - Closing Remarks
- Running NVIDIA Parabricks on DNAnexus
  - What is NVIDIA Parabricks?
  - Finding the Parabricks tools on DNAnexus
  - Running the Parabricks FQ-to-BAM Pipeline using the GUI
  - Running the Parabricks FQ-to-BAM Pipeline using the CLI
- Running NVIDIA Parabricks on GCP
  - What is NVIDIA Parabricks?
  - Starting a Compute Instance
  - Installing Parabricks
  - Testing Parabricks
  - Closing Remarks
- Running NVIDIA Parabricks on nf-core
  - What is NVIDIA Parabricks?
  - Finding the Parabricks modules on nf-core
  - Running the Parabricks modules on nf-core
- Running NVIDIA Parabricks on OCI
  - What is NVIDIA Parabricks?
  - Starting a GPU Instance for Parabricks Installation
  - Sign into the GPU instance to install and test Parabricks
  - Download and Install Parabricks
  - Closing Remarks
- Running NVIDIA Parabricks on Terra
  - What is NVIDIA Parabricks?
  - Running fq2bam on Terra

DeepVariant training using Parabricks

DeepVariant is a data analysis pipeline employing a deep neural network to identify genetic variants from next-generation DNA sequencing (NGS) data. While DeepVariant is exceptionally precise for various NGS data, there might be users keen on crafting tailored deep learning models meticulously suited for highly specific data.

The DeepVariant training pipeline has three major steps:

1. Run make_examples in “training” mode on the training and validation data sets,
2. Shuffle each set of examples and generate a data configuration file for each, and
3. Run model_train and model_eval.

Parabricks currently contains a GPU accelerated version of the first two steps.

Run make_examples in training mode

The "make_examples" step processes the input data, producing output suitable for use in subsequent steps. The output produced will include a label field.

Beginning with version 1.4.0, DeepVariant introduced an additional parameter in their WGS configuration through the `--channels "insert_size"` flag.

Depending on the nature of your data, you may wish to adjust the flags for the make_examples step, potentially leading to varying formats for the output examples. Please see the DeepVariant documentation for details regarding these options.

make_examples Quick Start

This code runs the "make_examples" step, combining the reference, BAM, VCF and BED files into a format suitable for use by the shuffle, model_train and model_eval steps.

```
# 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 \ --workdir /workdir \
nvcr.io/nvidia/clara/deepvariant_train:4.2.0-1 \ pbrun make_examples \ --ref
/workdir/${REFERENCE_FILE} \ --reads /workdir/${INPUT_BAM} \ --truth-variants
/workdir/${TRUTH_VCF} \ --confident-regions /workdir/${TRUTH_BED} \ --examples
/outputdir/${TFRECORD_FILE} \ --disable-use-window-selector-model \ --channel-insert-size
```

Compatible make_examples Baseline Command

```
( seq 0 $((N_SHARDS-1)) |\ parallel --halt 2 --line-buffer \ sudo docker run --volume
<INPUT_DIR>:/workdir --volume <OUTPUT_DIR>:/outputdir \
google/deepvariant:"${BIN_VERSION"} \ /opt/deepvariant/bin/make_examples \ --mode
training \ --ref "/workdir/${REF}" \ --reads "/workdir/${INPUT_BAM}" \ --examples
"/outputdir/validation_set.with_label.tfrecord@${N_SHARDS}.gz" \ --truth_variants
"/workdir/${TRUTH_VCF" \ --confident_regions
"/workdir/${TRUTH_BED}" \ --task {} \ --channels "insert_size" )
```

make_examples Reference

Run `deepvariant make_examples` in training mode to create tensorflow.Examples.

make_examples Input/Output file options

--ref REF

Genome reference to use. Must have an associated FAI index as well. Supports text or gzipped references. Should match the reference used to align the BAM file provided to --reads. (default: None)

Option is required.

--reads READS

Aligned, sorted, indexed BAM file containing the reads we want to call. Should be aligned to a reference genome compatible with --ref. (default: None)

Option is required.

--interval-file INTERVAL_FILE

Path to a BED file (.bed) for selective access. This option can be used multiple times. (default: None)

--confident-regions CONFIDENT_REGIONS

Regions that we are confident are hom-ref or a variant in BED format. Contig names must match those of the reference genome. (default: None)

Option is required.

--truth-variants TRUTH_VARIANTS

Tabix-indexed VCF file containing the truth variant calls for this labels which we use to label our examples. (default: None)

Option is required.

--examples EXAMPLES

Path to write tf.Example protos in TFRecord format. (default: None)

Option is required.

--proposed-variants PROPOSED_VARIANTS

Path of the vcf.gz file, which has proposed variants for the make examples stage. (default: None)

make_examples Tool Options:

--num-cpu-threads-per-stream NUM_CPU_THREADS_PER_STREAM

Number of CPU threads to use per stream. (default: 6)

--num-zipper-threads NUM_ZIPPER_THREADS

Number of threads for compression and writing output files. (default: 4)

--num-streams-per-gpu NUM_STREAMS_PER_GPU

Number of streams to use per GPU. (default: 2)

--disable-use-window-selector-model

Change the window selector model from Allele Count Linear to Variant Reads. This option will increase the accuracy and runtime. (default: None)

--gvcf

Generate variant calls in .gvcf Format. (default: None)

--norealign-reads

Do not locally realign reads before calling variants. Reads longer than 500 bp are never realigned. (default: None)

--sort-by-haplotypes

Reads are sorted by haplotypes (using HP tag). (default: None)

--keep-duplicates

Keep reads that are duplicate. (default: None)

--vsc-min-count-snps VSC_MIN_COUNT_SNPS

SNP alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-count-indels VSC_MIN_COUNT_INDELS

Indel alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-fraction-snps VSC_MIN_FRACTION_SNPS

SNP alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: 0.12)

--vsc-min-fraction-indels VSC_MIN_FRACTION_INDELS

Indel alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: None)

--min-mapping-quality MIN_MAPPING_QUALITY

By default, reads with any mapping quality are kept. Setting this field to a positive integer i will only keep reads that have a MAPQ >= i. Note this only applies to aligned reads. (default: 5)

--min-base-quality MIN_BASE_QUALITY

Minimum base quality. This option enforces a minimum base quality score for alternate alleles. Alternate alleles will only be considered if all bases in the allele have a quality greater than min_base_quality. (default: 10)

--mode MODE

Value can be one of [shortread, pacbio, ont]. By default, it is shortread. If mode is set to pacbio, the following defaults are used: --norealign-reads, --alt-aligned-pileup diff_channels, --vsc-min-fraction-indels 0.12. If mode is set to ont, the following defaults are used: -norealign-reads, --variant-caller VCF_CANDIDATE_IMPORTER. (default: shortread)

--alt-aligned-pileup ALT_ALIGNED_PILEUP

Value can be one of [none, diff_channels]. Include alignments of reads against each candidate alternate allele in the pileup image. (default: None)

--variant-caller VARIANT_CALLER

Value can be one of [VERY_SENSITIVE_CALLER, VCF_CANDIDATE_IMPORTER]. The caller to use to make examples. If you use VCF_CANDIDATE_IMPORTER, it implies force calling. Default is VERY_SENSITIVE_CALLER. (default: None)

--add-hp-channel

Add another channel to represent HP tags per read. (default: None)

--parse-sam-aux-fields

Auxiliary fields of the BAM/CRAM records are parsed. If either --sort-by-haplotypes or --add-hp-channel is set, then this option must also be set. (default: None)

--use-wes-model

If passed, the WES model file will be used. Only used in shortread mode. (default: None)

--run-partition

Divide the whole genome into multiple partitions and run multiple processes at the same time, each on one partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--include-med-dp

If True, include MED (default: None)

--normalize-reads

If True, allele counter left align INDELs for each read. (default: None)

--pileup-image-width PILEUP_IMAGE_WIDTH

Pileup image width. Only change this if you know your model supports this width. (default: 221)

--channel-insert-size

If True, add insert_size channel into pileup image. By default, this parameter is true in WGS and WES mode. (default: None)

--no-channel-insert-size

If True, don't add insert_size channel into the pileup image. (default: None)

--max-read-size-512

Allow deepvariant to run on reads of size 512bp. The default size is 320 bp. (default: None)

--prealign-helper-thread

Use an extra thread for the pre-align step. This parameter is more useful when --max-reads-size-512 is set. (default: None)

--max-reads-per-partition MAX_READS_PER_PARTITION

The maximum number of reads per partition that are considered before following processing such as sampling and realignment. (default: 1500)

--partition-size PARTITION_SIZE

The maximum number of basepairs allowed in a region before splitting it into multiple smaller subregions. (default: 1000)

--track-ref-reads

If True, allele counter keeps track of reads supporting ref. By default, allele counter keeps a simple count of the number of reads supporting ref. (default: None)

--phase-reads

Calculate phases and add HP tag to all reads automatically. (default: None)

--dbg-min-base-quality DBG_MIN_BASE_QUALITY

Minimum base quality in a k-mer sequence to consider. (default: 15)

--ws-min-windows-distance WS_MIN_WINDOWS_DISTANCE

Minimum distance between candidate windows for local assembly (default: 80)

--channel-gc-content

If True, add gc (default: None)

--channel-hmer-deletion-quality

If True, add hmer deletion quality channel into pileup image (default: None)

--channel-hmer-insertion-quality

If True, add hmer insertion quality channel into pileup image (default: None)

--channel-non-hmer-insertion-quality

If True, add non-hmer insertion quality channel into pileup image (default: None)

--skip-bq-channel

If True, ignore base quality channel. (default: None)

--aux-fields-to-keep AUX_FIELDS_TO_KEEP

Comma-delimited list of auxiliary BAM fields to keep. Values can be [HP, tp, t0] (default: HP)

--vsc-min-fraction-hmer-indels VSC_MIN_FRACTION_HMER_INDELS

Hmer Indel alleles occurring at least this be advanced as candidates. Use this threshold if hmer and non-hmer indels should be treated differently (Ultima reads)Default will use the same threshold for hmer and non-hmer indels, as defined in vsc_min_fraction_indels. (default: None)

--vsc-turn-on-non-hmer-ins-proxy-support

Add read-support from soft-clipped reads and other non-hmer insertion alleles,to the most frequent non-hmer insertion allele. (default: None)

--consider-strand-bias

If True, expect SB field in calls and write it to vcf (default: None)

--p-error P_ERROR

Basecalling error for reference confidence model. (default: 0.001)

--channel-ins-size

If true, add another channel to represent size of insertions. (good for flow-based sequencing) (default: None)

--max-ins-size MAX_INS_SIZE

Max insertion size for ins_size_channel, larger insertions will look like max (have max intensity) (default: 10)

--disable-group-variants

If using vcf_candidate_importer and multi-allelic sites are split across multiple lines in VCF, set to True so that variants are not grouped when transforming CallVariantsOutput to Variants. (default: None)

-L INTERVAL, --interval INTERVAL

Interval within which to call the variants from the BAM/CRAM file. Overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

Run shuffle

The shuffling of TensorFlow example data is a crucial stage in model training. In the DeepVariant training process the examples are globally shuffled as part of the preprocessing step.

This script shuffles TensorFlow records locally and in-memory.

shuffle Quick Start

```
# 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 \ --workdir /workdir \
nvcr.io/nvidia/clara/deepvariant_train:4.2.0-1 \ pbrun shuffle \ --input-pattern-list
/workdir/validation_set.with_label.tfrecord-?????-of-00016.gz \ --output-pattern-
prefix /outputdir/validation_set.with_label.shuffled \ --output-dataset-config-pbtxt
/outputdir/validation_set.dataset_config.pbtxt \ --output-dataset-name HG001 \ --
direct-num-workers 16
```

Compatible shuffle Baseline Command

```
python3 shuffle_tfrecords_lowmem.py \ --
input_pattern_list="${INPUT_DIR}/validation_set.with_label.tfrecord=?????-of-
00016.gz" \ --
output_pattern_prefix="${OUTPUT_DIR}/validation_set.with_label.shuffled" \ --
output_dataset_config="${OUTPUT_DIR}/validation_set.dataset_config.pbtxt" \ --
putput_dataset_name="HG001" \ --direct_num_workders=16 \ --step=1
```

shuffle Reference

Shuffle examples globally.

Shuffle Input/Output file options

--output-dataset-config-pbtxt OUTPUT_DATASET_CONFIG_PBTXT

Human-readable version of DeepVariantDatasetConfig. (default: None)

Option is required.

--input-pattern-list INPUT_PATTERN_LIST [INPUT_PATTERN_LIST ...]

TFRecord filename pattern. (default: None)

Option is required.

Shuffle Tool Options:

--output-pattern-prefix OUTPUT_PATTERN_PREFIX

Filename pattern for the output TFRecords. (default: None)

Option is required.

--output-dataset-name OUTPUT_DATASET_NAME

Option is required.

--direct-num-workers DIRECT_NUM_WORKERS

Number of writer threads (default: 1)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

Run model_train and model_eval

We provide a Jupyter Notebook with a more detailed example of re-training DeepVariant 1.5 using Parabricks and additional instructions on the *model_train* and *model_eval* steps.

See also the DeepVariant training documentation, and the original Shuffle program.

How-Tos

The 'How-To' sections go into more detail on how to use the NVIDIA Parabricks software using larger data sets, a wider variety of options, and different ways of analyzing the data.

We start with how to perform germline calling and somatic calling. Additional discussions on somatic, exome, and transcriptome calling will be available in the future.

- Whole-Genome Small Variant Calling
- Whole-Genome Somatic Small Variant Calling

Whole-Genome Small Variant Calling

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.

Whole-Genome Somatic Small Variant Calling

This how-to explores calling variants using Mutect2 in the SEQC2 Consortium Sample.

Software Prerequisites

- Samtools and BWA
- SRA Toolkit
- NVIDIA Parabricks
- GATK4 v4.2.0.0
- bedtools
- R (vcfR, UpSetR)

Compute Requirements and Configuration

- At least 512GB of disk space (preferably on “balanced” or SSD storage)
- At least two NVIDIA Tesla T4 GPUs; the commands and time estimates below utilize four NVIDIA Tesla T4 GPUs.

- At least 24 CPU cores; the commands below use 48 vCPU cores.
- At least 48GB of RAM per GPU
- A functional Parabricks install (version 4.0 or newer)

The examples below use an AWS VM g4dn.12xlarge with 48 vCPU cores, 192GB of RAM, and 4 NVIDIA Tesla T4 GPUs running Ubuntu 18.04.5 LTS.

Introduction

This how-to runs through a full Whole Genome Sequencing (WGS) somatic variant analysis pipeline for calling SNPs, MNPs, and small indels on real 30X short-read human data. Such analyses are commonly used in cancer genomics studies. For WGS somatic variant analysis, you will utilize the example data generated by "**The Somatic Mutation Working Group of the SEQC2 Consortium**". The dataset contains multiplatform sequencing from HCC1395, which is a triple negative breast cancer cell line, and HCC1395 BL, which is the matched normal cell line. The SEQC2 dataset contains whole genome (WGS) and Exome, sequencing performed at six different sequencing centers, and multiple replicates to minimized potential biases from sequencing technologies/assays. Furthermore, the dataset was processed through nine bioinformatics pipelines to evaluate accuracy and reproducibility. The SEQC2 consortium reports artifacts generated due to sample and library processing, along with evaluating the capabilities and limitations of bioinformatics tools for artifact detection and removal. A series of detailed articles are available below:

- Establishing reference samples for detection of somatic mutations and germline variants with NGS technologies
- Toward best practice in cancer mutation detection with whole-genome and whole-exome sequencing
- Robust Cancer Mutation Detection with Deep Learning Models Derived from Tumor-Normal Sequencing Data
- Whole genome and exome sequencing reference datasets from a multi-center and cross-platform benchmark study
- Personalized genome assembly for accurate cancer somatic mutation discovery using tumor-normal paired reference samples

- Comprehensive Assessment of Somatic Copy Number Variation Calling Using Next-Generation Sequencing Data

- SEQC2

The SEQC2 dataset is well characterized and the **"Truth Variant Call set"** provided by the consortium allows you to validate the results from the WGS somatic variant analysis pipeline.

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 Parabricks tools for quality control to assess the variant caller results. We don't cover annotation or filtering in this How-To.

Download Example FASTQ Files

The example dataset used for this how-to resides on NCBI SRA and can be downloaded using `wget`. Once the SRA files are downloaded, install the SRA Toolkit to convert SRA files to paired-end FASTQ files using the instructions given below. Note that SRR7890827 is a normal sample and SRR7890824 is a tumor sample from the HCC1395 cell line. Paired-end WGS sequencing was performed for each sample on an Illumina HiSeq X.

```
## Download publicly available SRA files using wget. Both files are # about 65 GB in size. # Normal sample $ wget https://sra-pub-run-odp.s3.amazonaws.com/sra/SRR7890827/SRR7890827 --output-document=SRR7890827.sra # Tumor sample $ wget https://sra-pub-run-odp.s3.amazonaws.com/sra/SRR7890824/SRR7890824 --output-document=SRR7890824.sra ## Convert SRA to FASTQ files $ fastq-dump --split-files ./SRR7890827.sra --gzip $ fastq-dump --split-files ./SRR7890824.sra --gzip
```

Once the SRA files have been converted to FASTQ format, they are no longer needed and may be deleted.

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, download and index a reference genome to use in alignment and variant calling. You'll use GRCh38 for alignment, which may be downloaded from this page:

- GDC Reference Files | NCI Genomic Data Commons

From that website, download and extract the contents of:

- GRCh38.d1.vd1.fa.tar.gz

If you don't want to build your own BWA indices, also download and extract the contents of:

- GRCh38.d1.vd1_BWA.tar.gz and
- GRCh38.d1.vd1_GATK_indices.tar.gz

Otherwise you'll need to build your own indices. Instructions for building the indices are given below.

You'll need these additional VCF files and their indices:

- dbSNP-version155 (GCF_000001405.39.gz)
- 1000 genome Phase 3
- 1000 genome Phase 3 index
- Mills_and_1000G_gold_standard.indels
- Mills_and_1000G_gold_standard.indels index

Additionally, you'll need BED and VCF files for the truth set:

- High confidence calls and high confidence region

- High confidence INDELs
- High confidence SNPs

Index Reference Genome

If you use a reference file for which the index is not available, you can index it using BWA. Indexing should take 30 to 60 minutes.

```
$ tar -xvzf GRCh38.d1.vd1.fa.tar.gz ## Note this is baseline bwa. $ bwa index
GRCh38.d1.vd1.fa
```

Step1: Aligning the Fastq Files to the Reference Genome

The pbrun fq2bam 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 eight NVIDIA GPUs. The pbrun fq2bam command also generates a BQSR report, which is used to improve the base qualities within the BAM files and is used by MuTect2 downstream for variant calling. Note that this how-to adds a custom read group to make the sample easier to identify. If the sample is later merged with others in downstream analysis, adding a custom read group will be essential when running alignment for matched tumor/normal or multi-sample analyses. You will also need to pass a set of known sites to generate the BQSR report:

- GCF_000001405.39.gz
- ALL.wgs.1000G_phase3.GRCh38.ncbi_remapper.20150424.shapeit2_indels.vcf.gz
- Mills_and_1000G_gold_standard.indels.b38.primary_assembly.vcf.gz

```
## Aligning Normal sample FASTQ file $ docker run \ --gpus all \ --workdir /workdir \ --
rm \ --volume $(pwd):/workdir \ --volume $(pwd):/outputdir \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bam \ --ref
/workdir/GRCh38.d1.vd1.fa \ --in-fq /workdir/SRR7890827_1.fastq.gz
/workdir/SRR7890827_2.fastq.gz
"@RG\tID:id_SRR7890827_rg1\tLB:lib1\tPL:bar\tSM:sm_SRR7890827\tPU:pu_SRR78908
```

```
\ --knownSites
/workdir/Mills_and_1000G_gold_standard.indels.b38.primary_assembly.vcf.gz \ --
knownSites /workdir/GCF_000001405.39.gz \ --knownSites
/workdir/ALL.wgs.1000G_phase3.GRCh38.ncbi_remapper.20150424.shapeit2_indels.vc
\ --out-recal-file /outputdir/SRR7890827-WGS_FD_N_BQSR_REPORT.txt \ --bwa-
options=-Y \ --out-bam /outputdir/SRR7890827-WGS_FD_N.bam ## Aligning Tumor
sample FASTQ file $ docker run \ --gpus all \ --workdir /workdir \ --rm \ --volume
$(pwd):/workdir \ --volume $(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun fq2bam \ --ref /workdir/GRCh38.d1.vd1.fa \ --in-fq
/workdir/SRR7890824_1.fastq.gz /workdir/SRR7890824_2.fastq.gz
"@RG\tID:id_SRR7890824_rg1\tLB:lib1\tPL:bar\tSM:sm_SRR7890824\tPU:pu_SRR78908
\ --knownSites
/workdir/Mills_and_1000G_gold_standard.indels.b38.primary_assembly.vcf.gz \ --
knownSites /workdir/GCF_000001405.39.gz \ --knownSites
/workdir/ALL.wgs.1000G_phase3.GRCh38.ncbi_remapper.20150424.shapeit2_indels.vc
\ --out-recal-file /outputdir/SRR7890824-WGS_FD_T_BQSR_REPORT.txt \ --bwa-
options=-Y \ --out-bam /outputdir/SRR7890824-WGS_FD_T.bam
```

The pbrun fq2bam command performs read alignment, sorting, duplicate marking and indexing; it should take about 50 minutes running on four NVIDIA Tesla T4 GPUs. The output of fq2bam is a BAM file, a BAI index, and a BQSR report file. These files will be used as inputs to the SNP/indel caller run in the next step.

Step2: Generating a Set of Small Variant Calls Using Parabricks Mutect2

Parabricks currently supports the DeepVariant, HaplotypeCaller and Mutect2 variant callers. Different callers use slightly different methods for calling variants, with trade-offs in sensitivity and specificity for each. This example will run pbrun mutectcaller (Mutect2) to generate a set of small variant calls:

```
$ docker run \ --gpus all \ --workdir /workdir \ --rm \ --volume $(pwd):/workdir \ --
volume $(pwd):/outputdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun
mutectcaller \ --ref /workdir/GRCh38.d1.vd1.fa \ --in-tumor-bam
/workdir/SRR7890824-WGS_FD_T.bam \ --in-normal-bam /workdir/SRR7890827-
WGS_FD_N.bam \ --in-tumor-recal-file /workdir/RR7890824-
```

```
WGS_FD_T_BQSR_REPORT.txt \ --in-normal-recal-file /workdir/SRR7890827-
WGS_FD_N_BQSR_REPORT.txt \ --out-vcf /outputdir/SRR7890824-SRR7890827-
WGS_FD.vcf \ --tumor-name sm_SRR7890824 \ --normal-name sm_SRR7890827
```

This command puts the called variants in `SRR7890824-SRR7890827-WGS_FD.vcf` and creates the `SRR7890824-SRR7890827-WGS_FD.vcf.stats` file, which will be used in subsequent steps.

Looking for variants of clinical interest, you might consider running other variant callers and taking the intersection of two or more callers to generate a highly specific callset. On the other hand, if you are less concerned with false positives but need a highly-sensitive callset, you could take the union of multiple callers. Such vote-based consensus approaches have been employed in numerous large studies to address the shortcomings of individual callers.

First run MuTect2, which is the most widely used variant caller developed at Broad. Parabricks provides an accelerated version of MuTect2 v4.2.0.0 that is 10-15X faster than the community version. MuTect2 should run in under 30 minutes on a four-T4 GPU system.

The results of MuTect2 are plain-text VCF files.

Step3: For Validating the SNV and MNV Variants with truth set

Next, process the Mutect2 VCF files to extract non-indel variants using the GATK4 SelectVariants tool, which makes it possible to select a subset of variants based on various criteria in order to facilitate certain analyses. This example analysis extracts only the SNV and MNV variants, excluding small indels for further downstream validation of the variant calls in comparison to truthset (see detailed instructions below):

```
$ java -jar gatk-4.2.0.0/gatk-package-4.2.0.0-local.jar FilterMutectCalls \ -O
SRR7890824-SRR7890827-WGS_FD.FilterMutectCalls.vcf \ -R GRCh38.d1.vd1.fa \ -V
SRR7890824-SRR7890827-WGS_FD.vcf
```

Use GATK4 FilterMutectCalls to Apply Filters to the Raw Output of Mutect2

Once you have the SNV/MNV `.vcf` file, use GATK4 FilterMutectCalls to apply filters to the raw output of Mutect2. Parameters contained in M2FiltersArgumentCollection are described here.

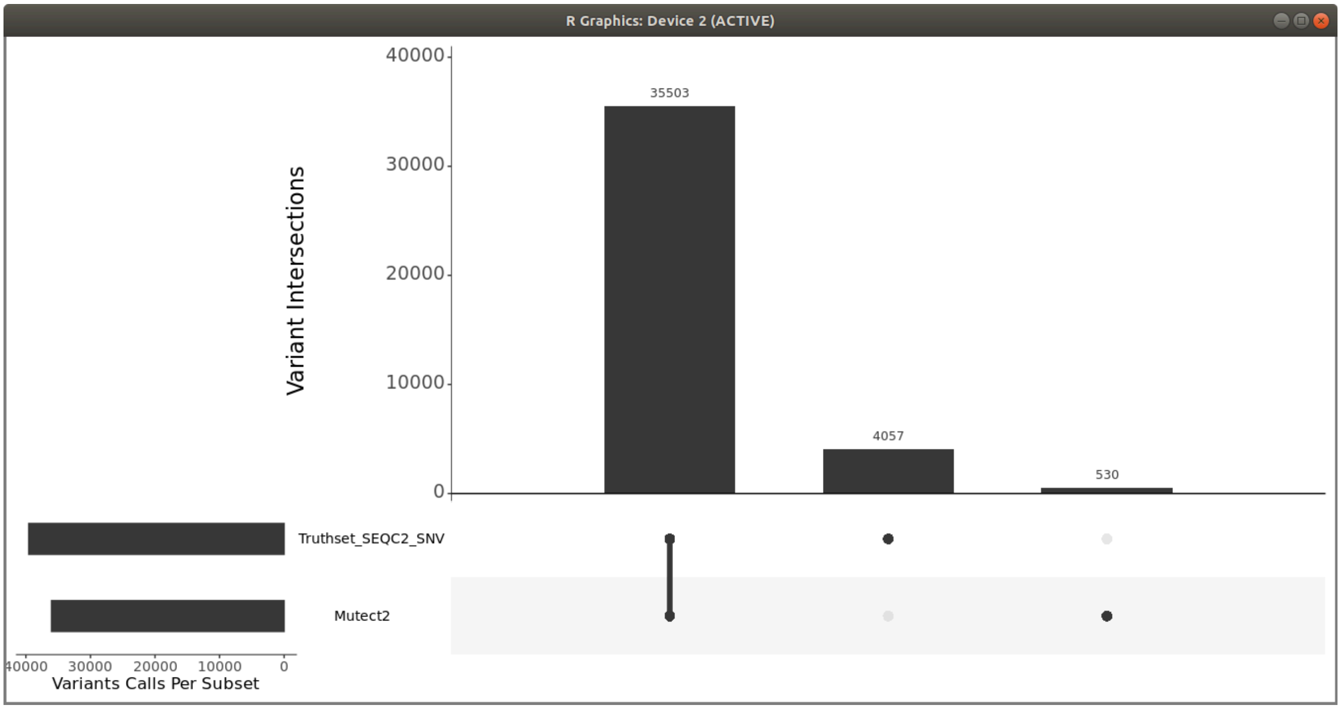
The following example uses a high confidence region bedfile to intersect the Mutect2 variants calls in the high confidence region and use the PASS filter variants for validation.

```
$ java -jar gatk-4.1.0.0/gatk-package-4.1.0.0-local.jar SelectVariants \ -R
GRCh38.d1.vd1.fa \ -V SRR7890824-SRR7890827-WGS_FD.FilterMutectCalls.vcf \ --
select-type-to-exclude INDEL \ -O SRR7890824-SRR7890827-WGS_FD.SNV-
MNV.FilterMutectCalls.vcf $ bedtools intersect \ -header \ -a SRR7890824-
SRR7890827-WGS_FD.SNV-MNV.FilterMutectCalls.vcf \ -b High-
Confidence_Regions_v1.2.bed > SRR7890824-SRR7890827-WGS_FD.SNV-
MNV.FilterMutectCalls.hc.vcf $ grep "#" SRR7890824-SRR7890827-WGS_FD.SNV-
MNV.FilterMutectCalls.hc.vcf > mutect_header.txt $ grep -v "#" SRR7890824-
SRR7890827-WGS_FD.SNV-MNV.FilterMutectCalls.hc.vcf | awk '{if ($7 ==
"PASS")print}' > mutect_body.txt $ cat mutect_header.txt mutect_body.txt >
SRR7890824-SRR7890827-WGS_FD.SNV-MNV.FilterMutectCalls.hc.PASS.vcf
```

Once you have the filtered Mutect2 calls, use R vcfR and upSetR packages to generate an upset plot to evaluate the performance of the variant calling pipeline. Use the following code to generate an upset plot:

```
> library(vcfR) > library(UpSetR) > ## SEQC2 Somatic SNV upset plot > ## Load the vcf
files from truthset and Mutect2 > Truthset_snv_hc <- read.vcfR("high-
confidence_sSNV_in_HC_regions_v1.2.hc.vcf.gz") > mutect_snv_hc <-
read.vcfR("SRR7890824-SRR7890827-WGS_FD.SNV-
MNV.FilterMutectCalls.hc.PASS.vcf") > ### extract chromosome and position to
compare > Truthset_snv_hc <- as.vector(paste(Truthset_snv_hc@fix[, "CHROM"],
Truthset_snv_hc@fix[, "POS"], sep = "_")) > mutect_snv_hc <-
as.vector(paste(mutect_snv_hc@fix[, "CHROM"], mutect_snv_hc@fix[, "POS"], sep =
"_")) > ## create the read set > read_sets <- list(Truthset_SEQC2_SNV =
Truthset_snv_hc, Mutect2 = mutect_snv_hc) > upset(fromList(read_sets), order.by =
c("freq"), keep.order = TRUE, sets = c("Mutect2", "Truthset_SEQC2_SNV"), point.size =
3.5, line.size = 2, mainbar.y.label = "Variant Intersections", text.scale = c(2, 2, 1.5, 1.5,
1.5, 1.5),sets.x.label = "Variants Calls Per Subset")
```

You should get a plot similar to the following:



You can also use *som.py* from the Illumina hap.py package to compare the two VCF files.

References and Useful Links

https://www.nvidia.com/en-us/clara/genomics/

https://sites.google.com/view/seqc2/home

https://gatk.broadinstitute.org/hc/en-us/articles/360036360312-Mutect2

https://github.com/Illumina/strelka/blob/v2.9.x/docs/userGuide/README.md

https://bioinformatics.mdanderson.org/public-software/muse/

https://csb5.github.io/lofreq/

https://github.com/genome/somatic-sniper/blob/master/gmt/documentation.md

https://github.com/Illumina/manta

https://github.com/Illumina/hap.py

https://cran.r-project.org/web/packages/vcfR/readme/README.html https://cran.r-project.org/web/packages/UpSetR/vignettes/basic.usage.html

Tool Reference

- applybqsr
- bam2fq
- bammetrics
- bamsort
- bqsr
- collectmultiplemetrics
- dbsnp
- deepsomatic
- deepvariant
- deepvariant_germline
- fq2bam (BWA-MEM + GATK)
- fq2bam_meth
- fq2bamfast (BWA-MEM + GATK)
- genotypegvcf
- germline (GATK Germline Pipeline)
- haplotypecaller
- indexgvcf
- markdup
- minimap2 (Beta)
- mutectcaller
- pacbio_germline (Beta)
- postpon
- prepon
- rna_fq2bam
- somatic (Somatic Variant Caller)
- starfusion

Tools By Category

Category	Tool
FASTQ/BAM Processing	• applybqsr • bam2fq

	• bamsort • bqsr • fq2bam (BWA-MEM + GATK) • fq2bamfast (BWA-MEM + GATK) • fq2bam_meth • markdup • minimap2 (Beta)
Variant Calling	• deepsomatic • deepvariant • deepvariant_germline • germline (GATK Germline Pipeline) • haplotypecaller • mutectcaller • pacbio_germline (Beta) • postpon • prepon • somatic (Somatic Variant Caller)
RNA	• rna_fq2bam • starfusion
Quality Control	• bammetrics • collectmultiplemetrics
Variant Processing	• dbsnp
GVCF Processing	• genotypegvcf • indexgvcf

applybqsr

Updates the Base Quality Scores using the BQSR report.

This tool recalibrates quality scores in a BAM file using the report generated by the bqsr [link] tool. This should be applied after alignment but before variant calling to maximize final accuracy in variant calling, as recommended by GATK best practices.

Please note that the **applybqsr** tool will use at most two GPUs.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun applybqsr \ --ref /workdir/${REFERENCE_FILE} \ --in-bam
/workdir/${INPUT_BAM} \ --in-recal-file /workdir/${INPUT_RECAL_FILE} \ --out-bam
/outputdir/${OUTPUT_BAM}
```

Compatible GATK4 Command

The command below is the GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command.

```
$ gatk ApplyBQSR \ --java-options -Xmx30g \ -R <INPUT_DIR>/${REFERENCE_FILE} \ -I
<INPUT_DIR>/${INPUT_BAM} \ --bqsr-recal-file <INPUT_DIR>/${INPUT_RECAL_FILE} \
-O <OUTPUT_DIR>/${OUTPUT_BAM}
```

applybqsr Reference

Update the Base Quality Scores using the BQSR report.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-bam IN_BAM

Path to the BAM file. (default: None)

Option is required.

--in-recal-file IN_RECAL_FILE

Path to the BQSR report file. (default: None)

Option is required.

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-bam OUT_BAM

Output BAM file. (default: None)

Option is required.

Tool Options:

-L INTERVAL, --interval INTERVAL

Interval within which to call applyBQSR from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

Performance Options:

--num-threads NUM_THREADS

Number of threads for worker. (default: 8)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

bam2fq

Run bam2fq to convert BAM/CRAM to FASTQ.

This tool un-aligns a BAM file, reversing it from BAM to FASTQ format. This can be useful if the BAM needs to be re-aligned to a newer or different reference genome by applying bam2fq followed by fq2bam (BWA-MEM + GATK) with the new reference genome.

For paired reads, bam2fq will append "/1" to the 1st read name, and "/2" to the 2nd read name.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun bam2fq \ --ref /workdir/${REFERENCE_FILE} \ --in-bam
/workdir/${INPUT_BAM} \ --out-prefix /workdir/${Prefix_for_output_fastq_files}
```

Compatible CPU-based BWA-MEM, GATK4 Commands

The command below is the bwa-0.7.15 and GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
$ gatk SamToFastq \ -I <INPUT_DIR>/${INPUT_BAM} \ -F
```

```
<OUTPUT_DIR>/${OUTPUT_FASTQ_1} \ -F2 <OUTPUT_DIR>/${OUTPUT_FASTQ_2}
```

bam2fq Reference

Run bam2fq to convert BAM/CRAM to FASTQ.

Input/Output file options

--ref REF

Path to the reference file. This argument is only required for CRAM input. (default: None)

--in-bam IN_BAM

Path to the input BAM/CRAM file to convert to fastq.gz. (default: None)

Option is required.

--out-prefix OUT_PREFIX

Prefix filename for output fastq files. (default: None)

Option is required.

Tool Options:

--out-suffixF OUT_SUFFIXF

Output suffix used for paired reads that are first in pair. The suffix must end with ".gz". (default: _1.fastq.gz)

--out-suffixF2 OUT_SUFFIXF2

Output suffix used for paired reads that are second in pair. The suffix must end with ".gz". (default: _2.fastq.gz)

--out-suffixO OUT_SUFFIXO

Output suffix used for orphan/unmatched reads that are first in pair. The suffix must end with ".gz". If no suffix is provided, these reads will be ignored. (default: None)

--out-suffixO2 OUT_SUFFIXO2

Output suffix used for orphan/unmatched reads that are second in pair. The suffix must end with ".gz". If no suffix is provided, these reads will be ignored. (default: None)

--out-suffixS OUT_SUFFIXS

Output suffix used for single-end/unpaired reads. The suffix must end with ".gz". If no suffix is provided, these reads will be ignored. (default: None)

--rg-tag RG_TAG

Split reads into different fastq files based on the read group tag. Must be either PU or ID. (default: None)

--remove-qc-failure

Remove reads from the output that have abstract QC failure. (default: None)

Performance Options:

--num-threads NUM_THREADS

Number of threads to run. (default: 8)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

bammetrics

Accelerated GATK4 CollectWGSMetrics.

This tool applies an accelerated version of the GATK CollectWGSMetrics for assessing coverage and quality of an aligned whole-genome BAM file. This includes metrics such as the fraction of reads that pass the base and mapping quality filters, and the coverage levels (read-depth) across the genome. These act as an overall quality check for the user, allowing assessment of how well a sequencing run has performed.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun bammetrics \ --ref /workdir/${REFERENCE_FILE} \ --bam /workdir/${INPUT_BAM} \ --out-metrics-file /outputdir/${METRICS_FILE}
```

Compatible GATK4 Command

The command below is the GATK4 counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command.

```
$ gatk CollectWgsMetrics \ -R <INPUT_DIR>/${REFERENCE_FILE} \ -I <INPUT_DIR>/${INPUT_BAM} \ -O <OUTPUT_DIR>/${METRICS_FILE}
```

bammetrics Reference

Run bammetrics on a BAM file to generate a metrics file.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--bam BAM

Path to the BAM file. (default: None)

Option is required.

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default:

None)
--out-metrics-file OUT_METRICS_FILE

Output Metrics File. (default: None)

Option is required.

Tool Options:

--minimum-base-quality MINIMUM_BASE_QUALITY

Minimum base quality for a base to contribute coverage. (default: 20)

--minimum-mapping-quality MINIMUM_MAPPING_QUALITY

Minimum mapping quality for a read to contribute coverage. (default: 20)

--count-unpaired

If true, count unpaired reads and paired reads with one end unmapped. (default: None)

--coverage-cap COVERAGE_CAP

Treat positions with coverage exceeding this value as if they had coverage at this value (but calculate the difference for PCT_EXC_CAPPED). (default: 250)

-L INTERVAL, --interval INTERVAL

Interval within which to collect metrics from the BAM/CRAM file. All intervals will have a padding of 0 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

Performance Options:

--num-threads NUM_THREADS

Number of threads to run. (default: 12)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

bamsort

Sort BAM files.

This tool can sort the reads within a BAM file in a variety of ways, including by position in the genome (coordinate) or read name (queryname). This enables compatibility with the requirements of different downstream tools.

Five sort modes are supported:

- coordinate (Picard-compatible)
- coordinate (fgbio-compatible)
- queryname (Picard-compatible)
- queryname (fgbio-compatible)
- template coordinate sort (fgbio-compatible)

Allowed values for **--sort-order** are as follows:

- coordinate [default]
- queryname
- templatecoordinate

Allowed values for **--sort-compatibility** are as follows:

- picard [default]
- fgbio

coordinate and *queryname* sorting can be done in either *picard* or *fgbio* mode. *templatecoordinate* can only be done in fgbio mode.

Quick Start

```
$ # 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun bamsort \ --ref /workdir/${REFERENCE_FILE} \ --in-bam
/workdir/${INPUT_BAM} \ --out-bam /outputdir/${OUTPUT_BAM} \ --sort-order
coordinate
```

Compatible Picard Command

The command below is the Picard counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command.

```
$ java -Xmx30g -jar picard.jar SortSam \ I=<INPUT_DIR>/${INPUT_BAM} \ O=
<OUTPUT_DIR>/${OUTPUT_BAM}
```

bamsort Reference

Sort BAM files. There are five modes: Coordinate sort (Picard-compatible), Coordinate sort (fgbio-compatible), queryname sort (Picard-compatible), queryname sort (fgbio-compatible), and template coordinate sort (fgbio- compatible).

Input/Output file options

--in-bam IN_BAM

Path of BAM/CRAM for sorting. This option is required. (default: None)

Option is required.

--out-bam OUT_BAM

Path of BAM/CRAM file after sorting. (default: None)

Option is required.

--ref REF

Path to the reference file. (default: None)

Option is required.

Pipeline Options:

--sort-order SORT_ORDER

Type of sort to be done. Possible values are {coordinate,queryname,templatecoordinate}. (default: coordinate)

--sort-compatibility SORT_COMPATIBILITY

Sort comparator compatibility to be used for compatibility with other tools. Possible values are {picard,fgbio}. TemplateCoordinate will only use fgbio. (default: picard)

Performance Options:

--num-zip-threads NUM_ZIP_THREADS

Number of CPUs to use for zipping BAM files in a run (default 16 for coordinate sorts and 10 otherwise). (default: None)

--num-sort-threads NUM_SORT_THREADS

Number of CPUs to use for sorting in a run (default 10 for coordinate sorts and 16 otherwise). (default: None)

--max-records-in-ram MAX_RECORDS_IN_RAM

Maximum number of records in RAM when using a queryname or template coordinate sort mode; lowering this number will decrease maximum memory usage. (default: 65000000)

--mem-limit MEM_LIMIT

Memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

bqsr

This tool generates a Base Quality Score Recalibration report, which can be applied by the applybqsr tool, to recalibrate the quality scores in a BAM file. This is applied as part of the recommended GATK best practices to maximize accuracy in variant calling.

Quick Start

```
$ # 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun bqsr \ --ref /workdir/${REFERENCE_FILE} \ --in-bam
/workdir/${INPUT_BAM} \ --knownSites /workdir/${KNOWN_SITES_FILE} \ --out-recal-
file /outputdir/${INPUT_RECAL_FILE} \
```

Compatible GATK4 Command

The command below is the GATK4 counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command.

```
$ gatk BaseRecalibrator \ --java-options -Xmx30g \ --input
<INPUT_DIR>/${INPUT_BAM} \ --output <OUTPUT_DIR>/${INPUT_RECAL_FILE} \ --
known-sites <INPUT_DIR>/${KNOWN_SITES_FILE} \ --reference
<INPUT_DIR>/${REFERENCE_FILE}
```

bqsr Reference

Run BQSR on a BAM file to generate a BQSR report.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-bam IN_BAM

Path to the BAM file. (default: None)

Option is required.

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

Option is required.

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-recal-file OUT_RECAL_FILE

Output Report File. (default: None)

Option is required.

Tool Options:

-L INTERVAL, --interval INTERVAL

Interval within which to call BQSR from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

collectmultiplemetrics

Run a GPU-accelerated version of GATK's CollectMultipleMetrics.

This tool applies an accelerated version of the GATK CollectMultipleMetrics for assessing BAM file metrics such as alignment success, quality score distributions, GC bias, and sequencing artifacts. This functions as a 'meta-metrics' tool that can run any combination of the available metrics tools in GATK to perform an overall assessment of how well a sequencing run has been performed. The available metrics tools (PROGRAMs) can be found in the reference section below.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun collectmultiplemetrics \ --ref /workdir/${REFERENCE_FILE} \ --bam /workdir/${INPUT_BAM} \ --out-qc-metrics-dir /outputdir/${OUTPUT_DIR}\ --gen-all-metrics
```

Compatible GATK4 Command

The command below is the GATK4 counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command.

```
$ gatk CollectMultipleMetrics \ --REFERENCE_SEQUENCE <INPUT_DIR>/${REFERENCE_FILE} \ -I <INPUT_DIR>/${INPUT_BAM} \ -O <OUTPUT_DIR>/${OUTPUT_DIR} \ --PROGRAM CollectAlignmentSummaryMetrics \ --PROGRAM CollectInsertSizeMetrics \ --PROGRAM QualityScoreDistribution \ --
```

```
PROGRAM MeanQualityByCycle \ --PROGRAM CollectBaseDistributionByCycle \ --
PROGRAM CollectGcBiasMetrics \ --PROGRAM CollectSequencingArtifactMetrics \ --
PROGRAM CollectQualityYieldMetrics
```

collectmultiplemetrics Reference

Run collectmultiplemetrics on a BAM file to generate files for multiple classes of metrics.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--bam BAM

Path to the BAM file. (default: None)

Option is required.

--out-qc-metrics-dir OUT_QC_METRICS_DIR

Output Directory to store results of each analysis.

(default: None)

Option is required.

Tool Options:

--gen-all-metrics

Generate QC for every analysis. (default: None)

--gen-alignment

Generate QC for alignment summary metric. (default: None)

--gen-quality-score

Generate QC for quality score distribution metric. (default: None)

--gen-insert-size

Generate QC for insert size metric. (default: None)

--gen-mean-quality-by-cycle

Generate QC for mean quality by cycle metric. (default: None)

--gen-base-distribution-by-cycle

Generate QC for base distribution by cycle metric. (default: None)

--gen-gc-bias

Prefix name used to generate detail and summary files for gc bias metric. (default: None)

--gen-seq-artifact

Generate QC for sequencing artifact metric. (default: None)

--gen-quality-yield

Generate QC for quality yield metric. (default: None)

Performance Options:

--bam-decompressor-threads BAM_DECOMPRESSOR_THREADS

Number of threads for BAM decompression. (default: 3)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

dbsnp

Annotate variants based on a variant database.

This tool annotates the variant calls within a VCF file using the dbSNP database. The dbSNP database is a public archive of genetic variant information, consisting of known variants and data on whether each of these are considered to be neutral polymorphisms, polymorphisms with associated phenotypes, or regions of no variation.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun dbsnp \ --in-vcf /workdir/${INPUT_VCF} \ --out-vcf
/outputdir/${OUTPUT_VCF} \ --in-dbsnp-file /workdir/${DBSNP_DATABASE}
```

dbsnp Reference

Annotate variants based on a dbSNP.

Input/Output file options

--in-vcf IN_VCF

Path to the input VCF file. (default: None)

Option is required.

--in-dbsnp-file IN_DBSNP_FILE

Path to the input DBSNP file in vcf.gz format, with its tabix index. (default: None)

Option is required.

--out-vcf OUT_VCF

Output annotated VCF file. (default: None)

Option is required.

Options specific to this tool

(none)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

deepsomatic

GPU-accelerated DeepSomatic.

What is DeepSomatic?

DeepSomatic builds on the deep learning-based variant caller DeepVariant. It processes aligned reads from tumor and normal samples (in BAM or CRAM format), generates pileup image tensors, classifies these tensors using a convolutional neural network, and outputs somatic variants in standard VCF or gVCF files.

DeepSomatic is designed for somatic variant calling using tumor-normal sequencing data.

Parabricks has enhanced Google DeepSomatic to leverage GPUs extensively. The Parabricks version of DeepSomatic operates similarly to other common command line tools: it accepts two BAM files and a reference file as inputs and generates variants in a VCF file as output.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun deepsomatic \ --ref /workdir/${REFERENCE_FILE} \ --in-
tumor-bam /workdir/${INPUT_TUMOR_BAM} \ --in-normal-bam
/workdir/${INPUT_NORMAL_BAM} \ --out-variants /outputdir/${OUTPUT_VCF}
```

Compatible Google DeepVariant Commands

The commands below are the Google counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command.

See the Output Comparison page for comparing the results.

```
docker run \ --interactve \ --tty \ --rm \ --volume ${INPUT_DIR}:${INPUT_DIR} \ --
volume ${OUTPUT_DIR}:${OUTPUT_DIR} \ --workdir /workdir
google/deepvariant:1.6.1 \ run_deepsomatic \ --ref ${REFERENCE_FILE} \ --
reads_tumor ${TUMOR_BAM} \ --reads_normal ${NORMAL_BAM} \ --
customized_model ${DEEPSOMATIC_WGS_MODEL_FILE} \ --output_vcf
${OUTPUT_VCF} \ --make_examples_extra_args
"ws_use_window_selector_model=true" \ --num_shards=$(nproc)
```

deepsomatic Reference

Run DeepSomatic to convert BAM/CRAM to VCF.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-tumor-bam IN_TUMOR_BAM

Path to the input tumor BAM/CRAM file for somatic variant calling. (default: None)

Option is required.

--in-normal-bam IN_NORMAL_BAM

Path to the input normal BAM/CRAM file for somatic variant calling. (default: None)

Option is required.

--interval-file INTERVAL_FILE

Path to a BED file (.bed) for selective access. This option can be used multiple times. (default: None)

--out-variants OUT_VARIANTS

Path of the vcf/g.vcf/g.vcf.gz file after variant calling. (default: None)

Option is required.

--pb-model-file PB_MODEL_FILE

Path to a non-default parabricks model file for deepsomatic. (default: None)

Tool Options:

--disable-use-window-selector-model

Change the window selector model from Allele Count Linear to Variant Reads. This option will increase the accuracy and runtime. (default: None)

--no-channel-insert-size

If True, don't add insert_size channel into the pileup image. (default: False)

-L INTERVAL, --interval INTERVAL

Interval within which to call the variants from the BAM/CRAM file. Overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

Performance Options:

--num-cpu-threads-per-stream NUM_CPU_THREADS_PER_STREAM

Number of CPU threads to use per stream. (default: 6)

--num-streams-per-gpu NUM_STREAMS_PER_GPU

Number of streams to use per GPU. (default: 2)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

deepvariant

Run a GPU-accelerated DeepVariant algorithm.

What is DeepVariant?

DeepVariant is a deep learning based variant caller developed by Google for germline variant calling of high-throughput sequencing data. It works by taking aligned sequencing reads in BAM/CRAM format and utilizes a convolutional neural network (CNN) to classify the locus into true underlying genomic variation or sequencing error. DeepVariant can therefore call single nucleotide variants (SNVs) and insertions/deletions (InDels) from sequencing data at high accuracy in germline samples.

Why DeepVariant?

DeepVariant's approach is able to detect variants that are often missed by traditional (for example Bayesian) variant callers, and is known to reduce false positives. It offers several advantages over similar tools, including its ability to detect a wide range of variants with high accuracy, its scalability for analyzing large datasets, and its open source availability. Additionally, its deep learning-based approach allows it to provide better support for different sequencing platforms, as it can be retrained to provide higher accuracy for specific protocols or research areas.

How should I use DeepVariant?

DeepVariant is designed for use as a germline variant caller that can apply different models trained for specific sample types (such as whole genome and whole exome samples) to yield higher accuracy results. DeepVariant can be deployed within NVIDIA's Parabricks software suite, which is designed for accelerated secondary analysis in genomics, bringing industry standard tools and workflows from CPU to GPU, and delivering the same results at up to 60x faster runtimes. A 30x whole genome can be run through DeepVariant in as little as 8 minutes on an NVIDIA DGX station, compared to 5 hours on a CPU instance (m5.24xlarge, 96 x vCPU). DeepVariant in Parabricks is used in the same way as other command line tools that users are familiar with: It takes a BAM/CRAM and the reference genome as inputs and produces the variants (a VCF file) as outputs. **Currently, DeepVariant is supported for V100 and newer GPUs out of the box.**

Note

In version 3.8 the *--run-partition* option was added, which can lead to a significant speed increase. However, using the *--run-partition*, *--proposed-variants*, and *--gvcf* options at the same time will lead to a substantial slowdown. A warning will be issued and the *--run-partition* option will be ignored.

Available Operating Modes

Parabricks DeepVariant can run in one of three operating modes:

1. shortread
2. PacBio
3. ONT

See the **--mode** option below.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun deepvariant \ --ref /workdir/${REFERENCE_FILE} \ --in-bam
/workdir/${INPUT_BAM} \ --out-variants /outputdir/${OUTPUT_VCF}
```

Compatible Google DeepVariant Commands

The commands below are the Google counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
sudo docker run \ --volume <INPUT_DIR>:/input \ --volume <OUTPUT_DIR>:/output \
google/deepvariant:1.6.1 \ /opt/deepvariant/bin/run_deepvariant \ --model_type
WGS \ --ref /input/${REFERENCE_FILE} \ --reads /input/${INPUT_BAM} \ --output_vcf
/output/${OUTPUT_VCF} \ --num_shards $(nproc) \ --make_examples_extra_args
"ws_use_window_selector_model=true"
```

Models for additional GPUs

Parabricks DeepVariant supports the following models:

1. Short-read WGS

2. Short-read WES

3. PacBio

4. ONT

DeepVariant models for T4, V100 and all other GPUs which are Ampere and above architecture ship with the software.

deepvariant Reference

Run DeepVariant to convert BAM/CRAM to VCF.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.
--in-bam IN_BAM

Path to the input BAM/CRAM file for variant calling. (default: None)

Option is required.

--interval-file INTERVAL_FILE

Path to a BED file (.bed) for selective access. This option can be used multiple times. (default: None)

--out-variants OUT_VARIANTS

Path of the vcf/g.vcf/g.vcf.gz file after variant calling. (default: None)

Option is required.

--pb-model-file PB_MODEL_FILE

Path to a non-default parabricks model file for deepvariant. (default: None)

--pb-model-dir PB_MODEL_DIR

Path to a non-default parabricks model dir that contains multiple engine files for one model (default: None)

--proposed-variants PROPOSED_VARIANTS

Path of the vcf.gz file, which has proposed variants for the make examples stage. (default: None)

Tool Options:

--disable-use-window-selector-model

Change the window selector model from Allele Count Linear to Variant Reads. This option will increase the accuracy and runtime. (default: None)

--gvcf

Generate variant calls in .gvcf Format. (default: None)

--norealign-reads

Do not locally realign reads before calling variants. Reads longer than 500 bp are never realigned. (default: None)

--sort-by-haplotypes

Reads are sorted by haplotypes (using HP tag). (default: None)

--keep-duplicates

Keep reads that are duplicate. (default: None)

--vsc-min-count-snps VSC_MIN_COUNT_SNPS

SNP alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-count-indels VSC_MIN_COUNT_INDELS

Indel alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-fraction-snps VSC_MIN_FRACTION_SNPS

SNP alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: 0.12)

--vsc-min-fraction-indels VSC_MIN_FRACTION_INDELS

Indel alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: None)

--min-mapping-quality MIN_MAPPING_QUALITY

By default, reads with any mapping quality are kept. Setting this field to a positive integer i will only keep reads that have a MAPQ >= i. Note this only applies to aligned reads. (default: 5)

--min-base-quality MIN_BASE_QUALITY

Minimum base quality. This option enforces a minimum base quality score for alternate alleles. Alternate alleles will only be considered if all bases in the allele have a quality greater than min_base_quality. (default: 10)

--mode MODE

Value can be one of [shortread, pacbio, ont]. By default, it is shortread. If mode is set to pacbio, the following defaults are used: --norealign-reads, --alt-aligned-pileup diff_channels, --vsc-min-fraction-indels 0.12. If mode is set to ont, the following defaults are used: -norealign-reads, --variant-caller VCF_CANDIDATE_IMPORTER. (default: shortread)

--alt-aligned-pileup ALT_ALIGNED_PILEUP

Value can be one of [none, diff_channels]. Include alignments of reads against each candidate alternate allele in the pileup image. (default: None)

--variant-caller VARIANT_CALLER

Value can be one of [VERY_SENSITIVE_CALLER, VCF_CANDIDATE_IMPORTER]. The caller to use to make examples. If you use VCF_CANDIDATE_IMPORTER, it implies force calling. Default is VERY_SENSITIVE_CALLER. (default: None)

--add-hp-channel

Add another channel to represent HP tags per read. (default: None)

--parse-sam-aux-fields

Auxiliary fields of the BAM/CRAM records are parsed. If either --sort-by-haplotypes or --add-hp-channel is set, then this option must also be set. (default: None)

--use-wes-model

If passed, the WES model file will be used. Only used in shortread mode. (default: None)

--include-med-dp

If True, include MED_DP in the output gVCF records. (default: None)

--normalize-reads

If True, allele counter left align INDELs for each read. (default: None)

--pileup-image-width PILEUP_IMAGE_WIDTH

Pileup image width. Only change this if you know your model supports this width. (default: 221)

--channel-insert-size

If True, add insert_size channel into pileup image. By default, this parameter is true in WGS and WES mode. (default: None)

--no-channel-insert-size

If True, don't add insert_size channel into the pileup image. (default: None)

--max-read-size-512

Allow deepvariant to run on reads of size 512bp. The default size is 320 bp. (default: None)

--prealign-helper-thread

Use an extra thread for the pre-align step. This parameter is more useful when --max-reads-size-512 is set. (default: None)

--track-ref-reads

If True, allele counter keeps track of reads supporting ref. By default, allele counter keeps a simple count of the number of reads supporting ref. (default: None)

--phase-reads

Calculate phases and add HP tag to all reads automatically. (default: None)

--dbg-min-base-quality DBG_MIN_BASE_QUALITY

Minimum base quality in a k-mer sequence to consider. (default: 15)

--ws-min-windows-distance WS_MIN_WINDOWS_DISTANCE

Minimum distance between candidate windows for local assembly (default: 80)

--channel-gc-content

If True, add gc_content channel into pileup image (default: None)

--channel-hmer-deletion-quality

If True, add hmer deletion quality channel into pileup image (default: None)

--channel-hmer-insertion-quality

If True, add hmer insertion quality channel into pileup image (default: None)

--channel-non-hmer-insertion-quality

If True, add non-hmer insertion quality channel into pileup image (default: None)

--skip-bq-channel

If True, ignore base quality channel. (default: None)

--aux-fields-to-keep AUX_FIELDS_TO_KEEP

Comma-delimited list of auxiliary BAM fields to keep. Values can be [HP, tp, t0] (default: HP)

--vsc-min-fraction-hmer-indels VSC_MIN_FRACTION_HMER_INDELS

Hmer Indel alleles occurring at least this be advanced as candidates. Use this threshold if hmer and non-hmer indels should be treated differently (Ultima reads)Default will use the same threshold for hmer and non-hmer indels, as defined in vsc_min_fraction_indels. (default: None)

--vsc-turn-on-non-hmer-ins-proxy-support

Add read-support from soft-clipped reads and other non-hmer insertion alleles,to the most frequent non-hmer insertion allele. (default: None)

--consider-strand-bias

If True, expect SB field in calls and write it to vcf (default: None)

--p-error P_ERROR

Basecalling error for reference confidence model. (default: 0.001)

--channel-ins-size

If true, add another channel to represent size of insertions. (good for flow-based sequencing) (default: None)

--max-ins-size MAX_INS_SIZE

Max insertion size for ins_size_channel, larger insertions will look like max (have max intensity) (default: 10)

--disable-group-variants

If using vcf_candidate_importer and multi-allelic sites are split across multiple lines in VCF, set to True so that variants are not grouped when transforming CallVariantsOutput to Variants. (default: None)

--filter-reads-too-long

Ignore all input bam reads with size > 512bp (default: None)

--haploid-contigs HAPLOID_CONTIGS

Optional list of non autosomal chromosomes. For all listed chromosomes HET probabilities are not considered. (default: None)

-L INTERVAL, --interval INTERVAL

Interval within which to call the variants from the BAM/CRAM file. Overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

Performance Options:

--num-cpu-threads-per-stream NUM_CPU_THREADS_PER_STREAM

Number of CPU threads to use per stream. (default: 6)

--num-streams-per-gpu NUM_STREAMS_PER_GPU

Number of streams to use per GPU. (default: 2)

--run-partition

Divide the whole genome into multiple partitions and run multiple processes at the same time, each on one partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--max-reads-per-partition MAX_READS_PER_PARTITION

The maximum number of reads per partition that are considered before following processing such as sampling and realignment. (default: 1500)

--partition-size PARTITION_SIZE

The maximum number of basepairs allowed in a region before splitting it into multiple smaller subregions. (default: 1000)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the

PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)
--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

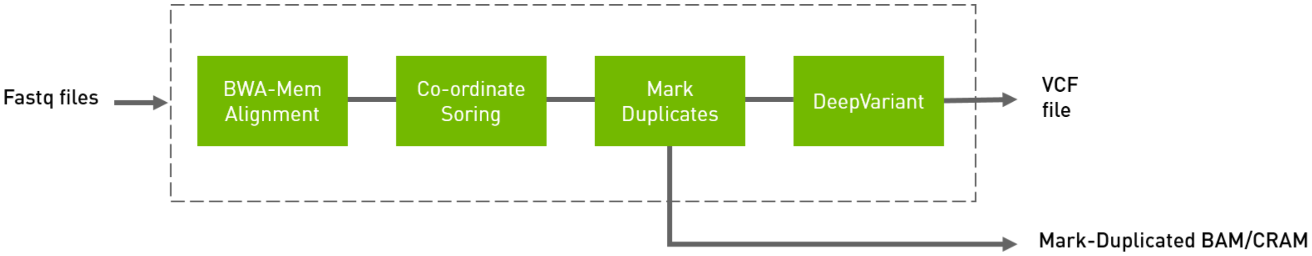
deepvariant_germline

Given one or more pairs of FASTQ files, you can run the germline variant tool to generate BAM, variants, duplicate metrics and recal.

The deepvariant germline tool includes alignment, sorting, and marking as well as the DeepVariant variant caller.

The inputs are BWA-indexed reference files and pair-ended FASTQ files. The outputs of this tool are the following:

- Aligned, co-ordinate sorted, duplicated marked BAM
- Variants in `vcf` / `g.vcf` / `g.vcf.gz` format



Quick Start

The following command runs the DeepVariant tool.

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun deepvariant_germline \ --ref /workdir/${REFERENCE_FILE}
\ --in-fq /workdir/${INPUT_FASTQ_1} /workdir/${INPUT_FASTQ_2} \ --out-variants
/outputdir/${OUTPUT_VCF_FILE}
```

Compatible Google DeepVariant Commands

The commands below are the Google counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# Run bwa-mem and pipe output to create sorted BAM $ bwa mem \ -t 32 \ -K
10000000 \ -R '@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \
<INPUT_DIR>/${REFERENCE_FILE} \ <INPUT_DIR>/${INPUT_FASTQ_1}
<INPUT_DIR>/${INPUT_FASTQ_2} | \ gatk SortSam \ --java-options -Xmx30g \ --
MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --SORT_ORDER
coordinate # Mark Duplicates $ gatk MarkDuplicates \ --java-options -Xmx30g \ -I
cpu.bam \ -O mark_dups_cpu.bam \ -M metrics.txt # Run deepvariant
BIN_VERSION="1.6.1" sudo docker run \ -v "${PWD}":"/input" \ -v
"${PWD}/output":"/output" \ -v "${PWD}/Ref":"/reference" \
google/deepvariant:"${BIN_VERSION}" \ /opt/deepvariant/bin/run_deepvariant \ --
model_type WGS \ --ref /reference/Homo_sapiens_assembly38.fasta \ --reads
/output/mark_dups_cpu.bam \ --output_vcf /output/"${OUTPUT_VCF_FILE}" \ --
```

```
num_shards $(nproc) \ --make_examples_extra_args
"ws_use_window_selector_model=true"
```

Models for additional GPUs

See the DeepVariant Models for additional GPUs section for instructions on downloading and using model files for additional GPUs.

deepvariant_germline Reference

Run the germline pipeline from FASTQ to VCF using a deep neural network analysis.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-fq [IN_FQ ...]

Path to the pair-ended FASTQ files followed by optional read groups with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The files must be in fastq or fastq.gz format. All sets of inputs should have a read group; otherwise, none should have a read group, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-fq [IN_SE_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq

or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-se-fq sampleX_1.fastq.gz --in-se-fq sampleX_2.fastq.gz . Example 2: --in-se-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-se-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)
--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--pb-model-file PB_MODEL_FILE

Path to a non-default parabricks model file for deepvariant. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of the report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of BAM file after Marking Duplicates. (default: None)

Option is required.

--out-variants OUT_VARIANTS

Path of the vcf/gvcf/gvcf.gz file after variant calling. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of a duplicate metrics file after Marking Duplicates. (default: None)

--proposed-variants PROPOSED_VARIANTS

Path of the VCF file, which has proposed variants for the make examples stage. (default: None)

Tool Options:

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000". (default: None)

--bwa-options BWA_OPTIONS

Pass supported bwa mem options as one string. The current original bwa mem supported options are -M, -Y and -T e.g. --bwa-options="-M -Y" (default: None)

--no-warnings

Suppress warning messages about system thread and memory usage. (default: None)

--filter-flag FILTER_FLAG

Don't generate SAM entries in the output if the entry's flag's meet this criteria. Criteria: (flag & filter != 0) (default: 0)

--skip-multiple-hits

Filter SAM entries whose length of SA is not 0. (default: None)

--min-read-length MIN_READ_LENGTH

Skip reads below minimum read length. They will not be part of the output. (default: None)

--align-only

Generate output BAM after bwa-mem. The output will not be co-ordinate sorted or duplicates will not be marked. (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--fix-mate

Add mate cigar (MC) and mate quality (MQ) tags to the output file. (default: None)

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--markdups-picard-version-2182

Assume marking duplicates to be similar to Picard version 2.18.2. (default: None)

--monitor-usage

Monitor approximate CPU utilization and host memory usage during execution. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. Ignored if --out-duplicate-metrics is not passed. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

--standalone-bqsr

Run standalone BQSR. (default: None)

--max-read-length-fq2bamfast MAX_READ_LENGTH_FQ2BAMFAST

Maximum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (Argument only applies to --fq2bamfast) (default: 480)

--min-read-length-fq2bamfast MIN_READ_LENGTH_FQ2BAMFAST

Minimum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (Argument only applies to --fq2bamfast) (default: 10)

--disable-use-window-selector-model

Change the window selector model from Allele Count Linear to Variant Reads. This option will increase the accuracy and runtime. (default: None)

--gvcf

Generate variant calls in .gvcf Format. (default: None)

--norealign-reads

Do not locally realign reads before calling variants. Reads longer than 500 bp are never realigned. (default: None)

--sort-by-haplotypes

Reads are sorted by haplotypes (using HP tag). (default: None)

--keep-duplicates

Keep reads that are duplicate. (default: None)

--vsc-min-count-snps VSC_MIN_COUNT_SNPS

SNP alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-count-indels VSC_MIN_COUNT_INDELS

Indel alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-fraction-snps VSC_MIN_FRACTION_SNPS

SNP alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: 0.12)

--vsc-min-fraction-indels VSC_MIN_FRACTION_INDELS

Indel alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: None)

--min-mapping-quality MIN_MAPPING_QUALITY

By default, reads with any mapping quality are kept. Setting this field to a positive integer i will only keep reads that have a MAPQ >= i. Note this only applies to aligned reads. (default: 5)

--min-base-quality MIN_BASE_QUALITY

Minimum base quality. This option enforces a minimum base quality score for alternate alleles. Alternate alleles will only be considered if all bases in the allele have a quality greater than min_base_quality. (default: 10)

--mode MODE

Value can be one of [shortread, pacbio, ont]. By default, it is shortread. If mode is set to pacbio, the following defaults are used: --norealign-reads, --alt-aligned-pileup diff_channels, --vsc-min-fraction-indels 0.12. If mode is set to ont, the following defaults are used: -norealign-reads, --variant-caller VCF_CANDIDATE_IMPORTER. (default: shortread)

--alt-aligned-pileup ALT_ALIGNED_PILEUP

Value can be one of [none, diff_channels]. Include alignments of reads against each candidate alternate allele in the pileup image. (default: None)

--variant-caller VARIANT_CALLER

Value can be one of [VERY_SENSITIVE_CALLER, VCF_CANDIDATE_IMPORTER]. The caller to use to make examples. If you use VCF_CANDIDATE_IMPORTER, it implies force calling. Default is VERY_SENSITIVE_CALLER. (default: None)

--add-hp-channel

Add another channel to represent HP tags per read. (default: None)

--parse-sam-aux-fields

Auxiliary fields of the BAM/CRAM records are parsed. If either --sort-by-haplotypes or --add-hp-channel is set, then this option must also be set. (default: None)

--use-wes-model

If passed, the WES model file will be used. Only used in shortread mode. (default: None)

--include-med-dp

If True, include MED_DP in the output gVCF records. (default: None)

--normalize-reads

If True, allele counter left align INDELs for each read. (default: None)

--pileup-image-width PILEUP_IMAGE_WIDTH

Pileup image width. Only change this if you know your model supports this width. (default: 221)

--channel-insert-size

If True, add insert_size channel into pileup image. By default, this parameter is true in WGS and WES mode. (default: None)

--no-channel-insert-size

If True, don't add insert_size channel into the pileup image. (default: None)

--max-read-size-512

Allow deepvariant to run on reads of size 512bp. The default size is 320 bp. (default: None)

--prealign-helper-thread

Use an extra thread for the pre-align step. This parameter is more useful when --max-reads-size-512 is set. (default: None)

--track-ref-reads

If True, allele counter keeps track of reads supporting ref. By default, allele counter keeps a simple count of the number of reads supporting ref. (default: None)

--phase-reads

Calculate phases and add HP tag to all reads automatically. (default: None)

--dbg-min-base-quality DBG_MIN_BASE_QUALITY

Minimum base quality in a k-mer sequence to consider. (default: 15)

--ws-min-windows-distance WS_MIN_WINDOWS_DISTANCE

Minimum distance between candidate windows for local assembly (default: 80)

--channel-gc-content

If True, add gc_content channel into pileup image (default: None)

--channel-hmer-deletion-quality

If True, add hmer deletion quality channel into pileup image (default: None)

--channel-hmer-insertion-quality

If True, add hmer insertion quality channel into pileup image (default: None)

--channel-non-hmer-insertion-quality

If True, add non-hmer insertion quality channel into pileup image (default: None)

--skip-bq-channel

If True, ignore base quality channel. (default: None)

--aux-fields-to-keep AUX_FIELDS_TO_KEEP

Comma-delimited list of auxiliary BAM fields to keep. Values can be [HP, tp, t0] (default: HP)

--vsc-min-fraction-hmer-indels VSC_MIN_FRACTION_HMER_INDELS

Hmer Indel alleles occurring at least this be advanced as candidates. Use this threshold if hmer and non-hmer indels should be treated differently (Ultima reads)Default will use the same threshold for hmer and non-hmer indels, as defined in vsc_min_fraction_indels. (default: None)

--vsc-turn-on-non-hmer-ins-proxy-support

Add read-support from soft-clipped reads and other non-hmer insertion alleles,to the most frequent non-hmer insertion allele. (default: None)

--consider-strand-bias

If True, expect SB field in calls and write it to vcf (default: None)

--p-error P_ERROR

Basecalling error for reference confidence model. (default: 0.001)

--channel-ins-size

If true, add another channel to represent size of insertions. (good for flow-based sequencing) (default: None)

--max-ins-size MAX_INS_SIZE

Max insertion size for ins_size_channel, larger insertions will look like max (have max intensity) (default: 10)

--disable-group-variants

If using vcf_candidate_importer and multi-allelic sites are split across multiple lines in VCF, set to True so that variants are not grouped when transforming CallVariantsOutput to Variants. (default: None)

--filter-reads-too-long

Ignore all input bam reads with size > 512bp (default: None)

--haploid-contigs HAPLOID_CONTIGS

Optional list of non autosomal chromosomes. For all listed chromosomes HET probabilities are not considered. (default: None)

Performance Options:

--fq2bamfast

Use fq2bamfast as the alignment tool instead of fq2bam (default: None)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting and marking. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--memory-limit MEMORY_LIMIT

System memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

--low-memory

Use low memory mode (default: None)

--num-cpu-threads-per-stage NUM_CPU_THREADS_PER_STAGE

Number of CPU threads to use per stage. (default: 8)

--bwa-nstreams BWA_NSTREAMS

Number of streams per GPU to use; note: more streams increases device memory usage (Argument only applies to --fq2bamfast) (default: 4)

--bwa-cpu-thread-pool BWA_CPU_THREAD_POOL

Number of threads to devote to CPU thread pool *per GPU* (Argument only applies to --fq2bamfast) (default: 16)

--num-cpu-threads-per-stream NUM_CPU_THREADS_PER_STREAM

Number of CPU threads to use per stream. (default: 6)

--num-streams-per-gpu NUM_STREAMS_PER_GPU

Number of streams to use per GPU. (default: 2)

--run-partition

Divide the whole genome into multiple partitions and run multiple processes at the same time, each on one partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--max-reads-per-partition MAX_READS_PER_PARTITION

The maximum number of reads per partition that are considered before following processing such as sampling and realignment. (default: 1500)

--partition-size PARTITION_SIZE

The maximum number of basepairs allowed in a region before splitting it into multiple smaller subregions. (default: 1000)

--read-from-tmp-dir

Running variant caller reading from bin files generated by Aligner and sort. Run postsort in parallel. This option will increase device memory usage. (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

> ⓘ **Note**
>
> The *--in-fq* option takes the names of two FASTQ files, optionally followed by a quoted read group. The FASTQ filenames must not start with a hyphen.

fq2bam (BWA-MEM + GATK)

Generate BAM/CRAM output given one or more pairs of FASTQ files. Can also optionally generate a BQSR report.

> ⓘ **Note**
>
> fq2bam will become an alias for fq2bamfast in the next major release. All fq2bam arguments will continue to be supported.

What is BWA-MEM?

BWA-MEM is a fast, accurate algorithm for mapping DNA sequence reads to a reference genome, performing local alignment and producing alignment for different parts of the query sequence. It is the default algorithm in Burrows-Wheeler Aligner (BWA) for reads that are longer than 70bp and is designed for high-throughput sequencing technologies such as Illumina and Pacific Biosciences.

Why BWA-MEM?

BWA-MEM is capable of handling longer reads and is less sensitive to errors than other alignment algorithms. It is therefore used for a variety of applications, from routine analysis of sequencing data to more advanced applications such as de novo assembly and variant calling.

Some of the advantages of using BWA-MEM over similar tools include:

1. It is faster than many other alignment algorithms, making it the ideal choice for high-throughput sequencing.

2. It has a lower false positive rate than many other alignment algorithms, which means fewer false-positive variants are reported.

3. It is memory-efficient, allowing it to be used on limited resources.

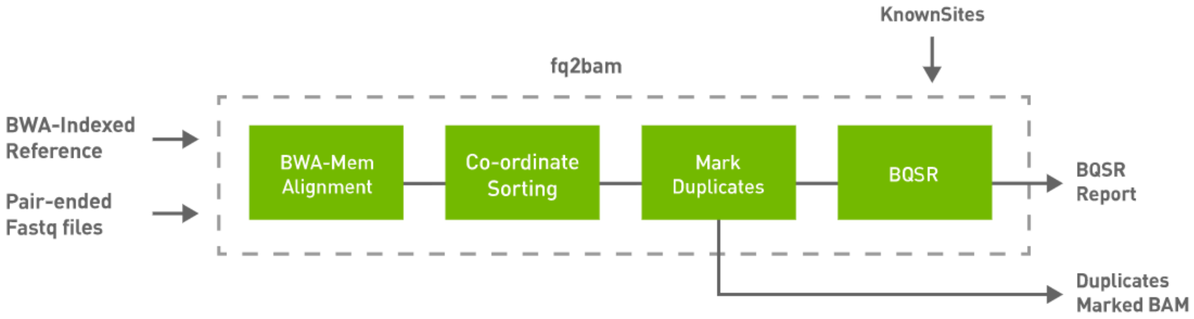
4. It is highly accurate, with a reported accuracy of over 99% on Illumina data.

What is fq2bam?

BWA-MEM can be deployed within Parabricks, a software suite designed for accelerated secondary analysis in genomics, bringing industry standard tools and workflows from CPU to GPU, and delivering the same results at up to 60x faster runtimes. FQ2BAM is the Parabricks wrapper for BWA-MEM, which will sort the output and can mark duplicates and recalibrate base quality scores in line with GATK best practices. A 30x whole genome can be run through FQ2BAM in as little as 17 minutes on an NVIDIA DGX system, compared to 4-9 hours on a CPU instance (m5.24xlarge, 96 x vCPU).

How should I use BWA-MEM in fq2bam?

fq2bam uses an accelerated version of BWA-MEM to generate BAM/CRAM output given one or more pairs of FASTQ files. The user can turn-off marking of duplicates by adding the *--no-markdups* option. The BQSR step is only performed if the *--knownSites input* and *--out-recal-file output* options are provided; doing so will also generate a BQSR report.



Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun fq2bam \ --ref /workdir/${REFERENCE_FILE} \ --in-fq
/workdir/${INPUT_FASTQ_1} /workdir/${INPUT_FASTQ_2} \ --knownSites
/workdir/${KNOWN_SITES_FILE} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-
recal-file /outputdir/${OUTPUT_RECAL_FILE}
```

Compatible CPU-based BWA-MEM, GATK4 Commands

The commands below are the bwa-0.7.15 and GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# Run bwa-mem and pipe the output to create a sorted BAM. $ bwa mem \ -t 32 \ -K
10000000 \ -R '@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \
<INPUT_DIR>/${REFERENCE_FILE} <INPUT_DIR>/${INPUT_FASTQ_1}
<INPUT_DIR>/${INPUT_FASTQ_2} | \ gatk SortSam \ --java-options -Xmx30g \ --
MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --SORT_ORDER
coordinate # Mark duplicates. $ gatk MarkDuplicates \ --java-options -Xmx30g \ -I
cpu.bam \ -O mark_dups_cpu.bam \ -M metrics.txt # Generate a BQSR report. $ gatk
BaseRecalibrator \ --java-options -Xmx30g \ --input mark_dups_cpu.bam \ --output
<OUTPUT_DIR>/${OUTPUT_RECAL_FILE} \ --known-sites
<INPUT_DIR>/${KNOWN_SITES_FILE} \ --reference <INPUT_DIR>/${REFERENCE_FILE}
```

fq2bam Reference

Run GPU-bwa mem, co-ordinate sorting, marking duplicates, and Base Quality Score Recalibration to convert FASTQ to BAM/CRAM.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-fq [IN_FQ ...]

Path to the pair-ended FASTQ files followed by optional read groups with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The files must be in fastq or fastq.gz format. All sets of inputs should have a read group; otherwise, none should have a read group, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-fq [IN_SE_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-se-fq sampleX_1.fastq.gz --in-se-fq sampleX_2.fastq.gz . Example 2: --in-se-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-se-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-fq-list IN_FQ_LIST

Path to a file that contains the locations of pair-ended FASTQ files. Each line must contain the location of two FASTQ files followed by a read group, each separated by a space. Each set of files (and associated read group) must be on a separate line. Files must be in fastq/fastq.gz format. Line syntax: <fastq_1> <fastq_2> <read group> (default: None)

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of a report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of a BAM/CRAM file. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of duplicate metrics file after Marking Duplicates. (default: None)

--out-qc-metrics-dir OUT_QC_METRICS_DIR

Path of the directory where QC metrics will be generated. (default: None)

Tool Options:

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000". (default: None)

--bwa-options BWA_OPTIONS

Pass supported bwa mem options as one string. The current original bwa mem supported options are -M, -Y and -T e.g. --bwa-options="-M -Y" (default: None)

--no-warnings

Suppress warning messages about system thread and memory usage. (default: None)

--filter-flag FILTER_FLAG

Don't generate SAM entries in the output if the entry's flag's meet this criteria. Criteria: (flag & filter != 0) (default: 0)

--skip-multiple-hits

Filter SAM entries whose length of SA is not 0. (default: None)

--min-read-length MIN_READ_LENGTH

Skip reads below minimum read length. They will not be part of the output. (default: None)

--align-only

Generate output BAM after bwa-mem. The output will not be co-ordinate sorted or duplicates will not be marked. (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--fix-mate

Add mate cigar (MC) and mate quality (MQ) tags to the output file. (default: None)

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--markdups-picard-version-2182

Assume marking duplicates to be similar to Picard version 2.18.2. (default: None)

--monitor-usage

Monitor approximate CPU utilization and host memory usage during execution. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. Ignored if --out-duplicate-metrics is not passed. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR. (default: None)

Performance Options:

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting and marking. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--memory-limit MEMORY_LIMIT

System memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

--low-memory

Use low memory mode (default: None)

--num-cpu-threads-per-stage NUM_CPU_THREADS_PER_STAGE

Number of CPU threads to use per stage. (default: 8)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

> ⓘ **Note**
>
> The *--in-fq* option takes the names of two FASTQ files, optionally followed by a quoted read group. The FASTQ filenames must not start with a hyphen.

> ⓘ **Note**
>
> When using the *--in-fq-list* option a read group is required on each line of the input file.

fq2bam_meth

Generate BAM/CRAM output given one or more pairs of FASTQ files from bisulfite sequencing (BS-Seq). Can also optionally generate a BQSR report.

What is fq2bam_meth?

The tool fq2bam_meth is a fast, accurate algorithm for mapping methylated DNA sequence reads to a reference genome, performing local alignment, and producing alignment for different parts of the query sequence. It implements the baseline tool bwa-meth [1] [2] in a performant method using fq2bamfast (BWA-MEM + GATK) as a backend for processing on GPU.

Why fq2bam_meth?

fq2bam_meth is the Parabricks wrapper for bwa-meth, which will sort the output and can mark duplicates and recalibrate base quality scores in line with GATK best practices.

The Parabricks fq2bam_meth tool is capable of handling longer reads and is less sensitive to errors than other alignment algorithms. We enable fast and accurate whole-genome bisulfite sequencing (WGBS) to detect DNA-methylation at the single base pair level [3].

Some of the advantages of using fq2bam_meth over similar tools include:

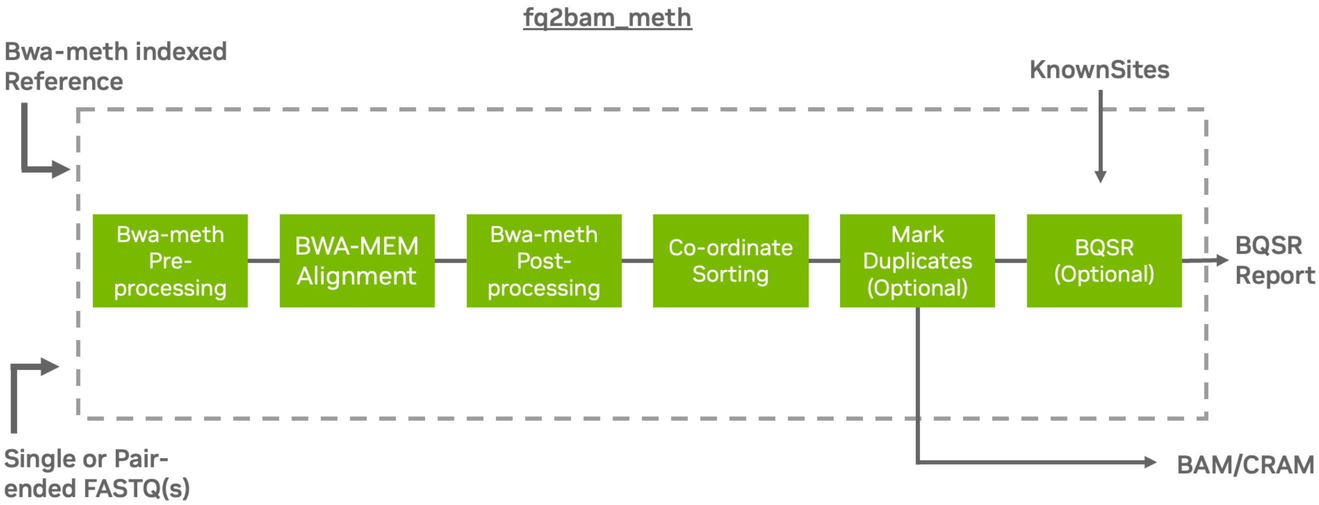
1. It is faster than many other BS-Seq alignment algorithms, making it the ideal choice for high-throughput analysis.

2. It maintains compatibility with existing CPU-based tools.

How should I use fq2bam_meth?

fq2bam_meth uses an accelerated version of BWA-MEM to generate BAM/CRAM output given one or more pairs of FASTQ files from BS-Seq. The user can turn-off marking of duplicates by adding the `--no-markdups` option. The BQSR step is only performed if the `--knownSites input` and `--out-recal-file output` options are provided; doing so will also generate a BQSR report.

Prior to running alignment, the reference genome must be converted using baseline bwa-meth. The bwa-meth indexing step produces a reference `fasta` file with a name formatted as `fasta.bwameth.c2t`. The indexing preparation step requires running `bwameth.py index $REF.fasta`. Baseline bwa-meth requires baseline BWA-MEM to be in the user's path for indexing functionality. Note that indexing is a time-consuming prerequisite that should only need to be completed once per reference genome. The `bwameth.py` script can be found here.



Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun fq2bam_meth \ --ref /workdir/${REFERENCE_FILE} \ --in-fq
/workdir/${INPUT_FASTQ_1} /workdir/${INPUT_FASTQ_2} \ --knownSites
/workdir/${KNOWN_SITES_FILE} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-
recal-file /outputdir/${OUTPUT_RECAL_FILE}
```

Compatible CPU-based bwa-meth, GATK4 Commands

The commands below are the bwa-meth-0.2.7, bwa-0.7.15, and GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

ⓘ **Note**

Set `--bwa-options="-K 10000000"` in fq2bam_meth and `-K 10000000` in baseline to produce compatible pair-ended results.

Note

fq2bam_meth will not strip `_R1` and `_R2` from read names during preprocessing like baseline bwa-meth.

```
# Run bwa-meth and pipe the output to create a sorted BAM. $ python bwa-meth.py \ -
-read-group '@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \ --
reference <INPUT_DIR>/${REFERENCE_FILE} <INPUT_DIR>/${INPUT_FASTQ_1}
<INPUT_DIR>/${INPUT_FASTQ_2} \ -t 32 -K 10000000 | \ gatk SortSam \ --java-
options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --
SORT_ORDER coordinate # Mark duplicates. $ gatk MarkDuplicates \ --java-options -
Xmx30g \ -I cpu.bam \ -O mark_dups_cpu.bam \ -M metrics.txt # Generate a BQSR
report. $ gatk BaseRecalibrator \ --java-options -Xmx30g \ --input
mark_dups_cpu.bam \ --output <OUTPUT_DIR>/${OUTPUT_RECAL_FILE} \ --known-
sites <INPUT_DIR>/${KNOWN_SITES_FILE} \ --reference
<INPUT_DIR>/${REFERENCE_FILE}
```

fq2bam_meth Reference

Run GPU-accelerated bwa-meth compatible alignment, co-ordinate sorting, marking duplicates, and Base Quality Score Recalibration to convert bisulfite reads from FASTQ to BAM/CRAM.

Input/Output file options

--ref REF

Path to the reference file. We will automatically look for <filename>.bwameth.c2t. Converted fasta reference must exist from prior conversion with baseline bwa-meth (default: None)

Option is required.

--in-fq [IN_FQ ...]

Path to the pair-ended FASTQ files followed by optional read groups with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The files must be in fastq or fastq.gz format. All sets of inputs should have a read group; otherwise, none should have a read group, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-fq [IN_SE_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-se-fq sampleX_1.fastq.gz --in-se-fq sampleX_2.fastq.gz . Example 2: --in-se-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-se-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-fq-list IN_FQ_LIST

Path to a file that contains the locations of pair-ended FASTQ files. Each line must contain the location of two FASTQ files followed by a read group, each separated by a space. Each set of files (and associated read group) must be on a separate line. Files must be in fastq/fastq.gz format. Line syntax: <fastq_1> <fastq_2> <read group> (default: None)

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of a report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of a BAM/CRAM file. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of duplicate metrics file after Marking Duplicates. (default: None)

--out-qc-metrics-dir OUT_QC_METRICS_DIR

Path of the directory where QC metrics will be generated. (default: None)

Tool Options:

--max-read-length MAX_READ_LENGTH

Maximum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (default: 480)

--min-read-length MIN_READ_LENGTH

Minimum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (default: 10)

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

--bwa-options BWA_OPTIONS

Pass supported bwa mem options as one string. The current original bwa mem supported options are -M, -Y and -T (e.g. --bwa-options="-M -Y") (default: None)

--no-warnings

Suppress warning messages about system thread and memory usage. (default: None)

--filter-flag FILTER_FLAG

Don't generate SAM entries in the output if the entry's flag's meet this criteria. Criteria: (flag & filter != 0) (default: 0)

--skip-multiple-hits

Filter SAM entries whose length of SA is not 0 (default: None)

--align-only

Generate output BAM after bwa-mem. The output will not be co-ordinate sorted or duplicates will not be marked (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--fix-mate

Add mate cigar (MC) and mate quality (MQ) tags to the output file. (default: None)

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--markdups-picard-version-2182

Assume marking duplicates to be similar to Picard version 2.18.2. (default: None)

--monitor-usage

Monitor approximate CPU utilization and host memory usage during execution. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. Ignored if --out-duplicate-metrics is not passed. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR. (default: None)

--set-as-failed SET_AS_FAILED

Flag alignments to strand 'f' or 'r' as failing quality-control (QC) with the failed QC flag 0x200. BS-Seq libraries are often to a single strand; other strands can be flagged as QC failures. Note: f == OT, r == OB. Valid options are 'f' or 'r' (default: None)

--do-not-penalize-chimeras

Turn off the default heuristic which marks alignments as failing QC if the longest match is less than 44% of the original sequence length. Alignments which fail this heuristic are also un-paired (default: None)

Performance Options:

--bwa-nstreams BWA_NSTREAMS

Number of streams per GPU to use; note: more streams increases device memory usage (default: 4)

--bwa-cpu-thread-pool BWA_CPU_THREAD_POOL

Number of threads to devote to CPU thread pool *per GPU* (default: 16)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting and marking. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --

gpuwrite. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)
--memory-limit MEMORY_LIMIT

System memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

--low-memory

Use low memory mode; will lower the number of streams per GPU (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

ⓘ **Note**

The *--in-fq* option takes the names of two FASTQ files, optionally followed by a quoted read group. The FASTQ filenames must not start with a hyphen.

ⓘ **Note**

When using the *--in-fq-list* option a read group is required on each line of the input file.

[1]

Baseline bwa-meth: https://github.com/brentp/bwa-meth/

[2]

Bwa-meth manuscript: http://arxiv.org/abs/1401.1129

[3]

https://doi.org/10.1038/s41587-022-01336-9

fq2bamfast (BWA-MEM + GATK)

Generate BAM/CRAM output given one or more pairs of FASTQ files. Can also optionally generate a BQSR report.

ⓘ **Note**

fq2bam will become an alias for fq2bamfast in the next major release.

What is BWA-MEM?

BWA-MEM is a fast, accurate algorithm for mapping DNA sequence reads to a reference genome, performing local alignment and producing alignment for different parts of the query sequence. It is the default algorithm in Burrows-Wheeler Aligner (BWA) for reads that are longer than 70bp and is designed for high-throughput sequencing technologies such as Illumina and Pacific Biosciences.

Why BWA-MEM?

BWA-MEM is capable of handling longer reads and is less sensitive to errors than other alignment algorithms. It is therefore used for a variety of applications, from routine analysis of sequencing data to more advanced applications such as de novo assembly and variant calling.

Some of the advantages of using BWA-MEM over similar tools include:

1. It is faster than many other alignment algorithms, making it the ideal choice for high-throughput sequencing.

2. It has a lower false positive rate than many other alignment algorithms, which means fewer false-positive variants are reported.

3. It is memory-efficient, allowing it to be used on limited resources.

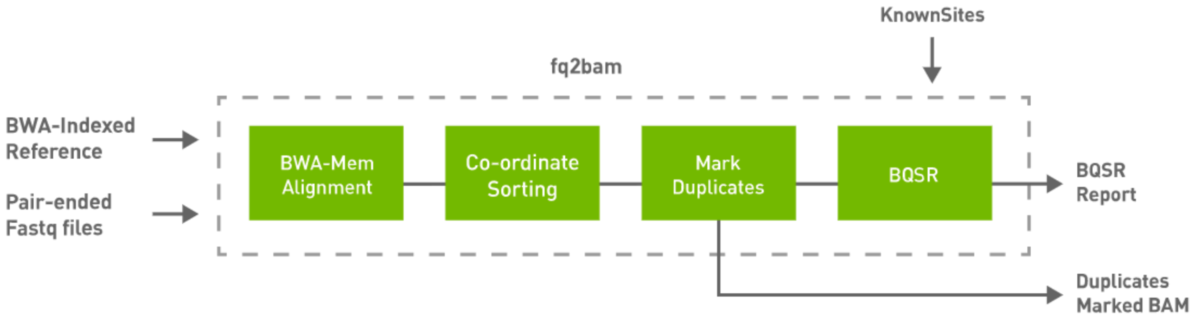
4. It is highly accurate, with a reported accuracy of over 99% on Illumina data.

What is fq2bamfast?

The tool fq2bamfast is Parabrick's new version of fq2bam's BWA-MEM implementation optimized for performance. We have kept the same command-line interface with small changes to support new performance options. Some BWA options may not yet be supported. Generally, fq2bamfast will use more device memory than fq2bam as a trade-off for better performance. If device memory is less than 50GB, one may need to experiment with the `--bwa-nstreams` or `--low-memory`.

How should I use BWA-MEM in fq2bamfast?

fq2bamfast uses an accelerated version of BWA-MEM to generate BAM/CRAM output given one or more pairs of FASTQ files. The user can turn-off marking of duplicates by adding the *--no-markdups* option. The BQSR step is only performed if the *--knownSites input* and *--out-recal-file output* options are provided; doing so will also generate a BQSR report.



Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun fq2bamfast \ --ref /workdir/${REFERENCE_FILE} \ --in-fq /workdir/${INPUT_FASTQ_1} /workdir/${INPUT_FASTQ_2} \ --knownSites /workdir/${KNOWN_SITES_FILE} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-recal-file /outputdir/${OUTPUT_RECAL_FILE}
```

Compatible CPU-based BWA-MEM, GATK4 Commands

The commands below are the bwa-0.7.15 and GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

> ⓘ **Note**
>
> Set `--bwa-options="-K 10000000"` to produce compatible pair-ended results.

```
# Run bwa-mem and pipe the output to create a sorted BAM. $ bwa mem \ -t 32 \ -K 10000000 \ -R '@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \ <INPUT_DIR>/${REFERENCE_FILE} <INPUT_DIR>/${INPUT_FASTQ_1} <INPUT_DIR>/${INPUT_FASTQ_2} | \ gatk SortSam \ --java-options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --SORT_ORDER coordinate # Mark duplicates. $ gatk MarkDuplicates \ --java-options -Xmx30g \ -I cpu.bam \ -O mark_dups_cpu.bam \ -M metrics.txt # Generate a BQSR report. $ gatk BaseRecalibrator \ --java-options -Xmx30g \ --input mark_dups_cpu.bam \ --output <OUTPUT_DIR>/${OUTPUT_RECAL_FILE} \ --known-sites <INPUT_DIR>/${KNOWN_SITES_FILE} \ --reference <INPUT_DIR>/${REFERENCE_FILE}
```

fq2bamfast Reference

Run GPU-bwa mem, co-ordinate sorting, marking duplicates, and Base Quality Score Recalibration to convert FASTQ to BAM/CRAM.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-fq [IN_FQ ...]

Path to the pair-ended FASTQ files followed by optional read groups with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The files must be in fastq or fastq.gz format. All sets of inputs should have a read group; otherwise, none should have a read group, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-fq [IN_SE_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-se-fq sampleX_1.fastq.gz --in-se-fq sampleX_2.fastq.gz . Example 2: --in-se-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-se-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-fq-list IN_FQ_LIST

Path to a file that contains the locations of pair-ended FASTQ files. Each line must contain the location of two FASTQ files followed by a read group, each separated by a space. Each set of files (and associated read group) must be on a separate line. Files must be in fastq/fastq.gz format. Line syntax: <fastq_1> <fastq_2> <read group> (default: None)

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of a report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of a BAM/CRAM file. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of duplicate metrics file after Marking Duplicates. (default: None)

--out-qc-metrics-dir OUT_QC_METRICS_DIR

Path of the directory where QC metrics will be generated. (default: None)

Tool Options:

--max-read-length MAX_READ_LENGTH

Maximum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (default: 480)

--min-read-length MIN_READ_LENGTH

Minimum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (default: 10)

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

--bwa-options BWA_OPTIONS

Pass supported bwa mem options as one string. The current original bwa mem supported options are -M, -Y and -T (e.g. --bwa-options="-M -Y") (default: None)

--no-warnings

Suppress warning messages about system thread and memory usage. (default: None)

--filter-flag FILTER_FLAG

Don't generate SAM entries in the output if the entry's flag's meet this criteria. Criteria: (flag & filter != 0) (default: 0)

--skip-multiple-hits

Filter SAM entries whose length of SA is not 0 (default: None)

--align-only

Generate output BAM after bwa-mem. The output will not be co-ordinate sorted or duplicates will not be marked (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--fix-mate

Add mate cigar (MC) and mate quality (MQ) tags to the output file. (default: None)

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--markdups-picard-version-2182

Assume marking duplicates to be similar to Picard version 2.18.2. (default: None)

--monitor-usage

Monitor approximate CPU utilization and host memory usage during execution. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. Ignored if --out-duplicate-metrics is not passed. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR. (default: None)

Performance Options:

--bwa-nstreams BWA_NSTREAMS

Number of streams per GPU to use; note: more streams increases device memory usage (default: 4)

--bwa-cpu-thread-pool BWA_CPU_THREAD_POOL

Number of threads to devote to CPU thread pool *per GPU* (default: 16)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting and marking. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--memory-limit MEMORY_LIMIT

System memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

--low-memory

Use low memory mode; will lower the number of streams per GPU (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

ⓘ **Note**

The *--in-fq* option takes the names of two FASTQ files, optionally followed by a quoted read group. The FASTQ filenames must not start with a hyphen.

ⓘ **Note**

When using the *--in-fq-list* option a read group is required on each line of the input file.

genotypegvcf

This tool converts variant calls in g.vcf format to VCF format.

This tool applies an accelerated GATK GenotypeGVCFs for joint genotyping, converting from `g.vcf` format to regular VCF format. This utilizes the HaplotypeCaller genotype likelihoods, produced with the `-ERC` GVCF flag, to joint genotype on one or more (multi-sample) `g.vcf` files.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun genotypegvcf \ --ref /workdir/${REFERENCE_FILE} \ --in-gvcf /workdir/${INPUT_GVCF_FILE} \ --out-vcf /outputdir/${OUTPUT_VCF}
```

Compatible CPU GATK4 Command

```
$ gatk GenotypeGVCFs \ -R <INPUT_DIR>/${REFERENCE_FILE} \ -V <INPUT_DIR>/${INPUT_GVCF_FILE} \ -O <OUTPUT_DIR>/${OUTPUT_VCF}
```

genotypegvcf Reference

Convert GVCF to VCF.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-gvcf IN_GVCF

Input a g.vcf or g.vcf.gz file that will be converted to VCF. Required. (default: None)

Option is required.

--out-vcf OUT_VCF

Path to output VCF file. (default: None)

Option is required.

Options specific to this tool

(none)

Performance Options:

--num-threads NUM_THREADS

Number of threads for worker. (default: 4)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

germline (GATK Germline Pipeline)

What is GATK?

GATK, the Genome Analysis Toolkit, is an industry standard software package developed by the Broad Institute of MIT and Harvard and designed to be used for a wide range of genomic analyses, including variant discovery, genotyping, and more. GATK is one of the most popular tools used in bioinformatics for analyzing next-generation sequencing datasets and is an industry standard for calling single nucleotide variants (SNVs) and insertions/deletions (InDels) from sequencing data in germline samples.

Why GATK?

GATK offers robust, accurate analysis of sequencing data and is frequently updated to include the latest best practices for variant discovery. With high reliability and the ability to be used for a number of use cases, GATK is a gold standard tool for any researcher working with next-generation sequencing data.

How should I use GATK?

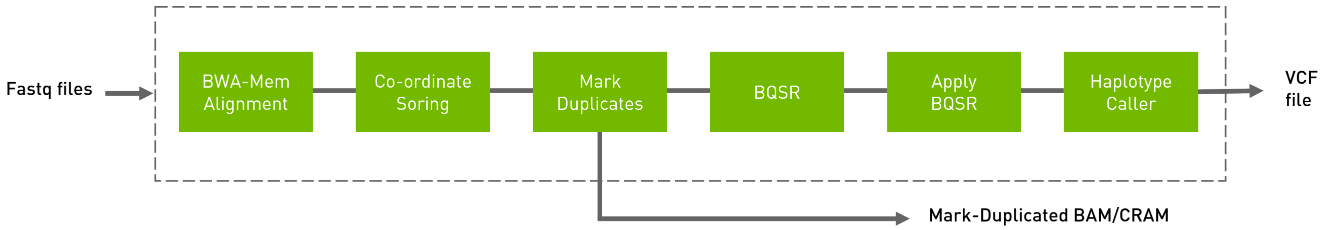
The GATK germline workflow for variant calling can be deployed within NVIDIA's Parabricks software suite, which is designed for accelerated secondary analysis in genomics, bringing industry standard tools and workflows from CPU to GPU and delivering the same results at up to 60x faster runtimes. A 30x whole genome can be analyzed in under 25 minutes on an NVIDIA DGX system, compared to over 30 hours on a CPU instance (m5.24xlarge, 96 x vCPU), and exomes can be analyzed in just 4 minutes. This means Parabricks, running on one NVIDIA DGX A100, can analyze up to 25,000 whole genomes per year. The NVIDIA team collaborated with the GATK team at the Broad Institute to evaluate the accuracy of germline workflows. Through this rigorous process, they verified that the Parabricks workflows produce results that are functionally equivalent to the CPU-native GATK versions.

As a specific example, benchmarking on publicly available Genome in a Bottle (GIAB) samples with the fq2bam and germline caller workflows from the Parabricks suite produced variant calling results that were >0.9999 equivalent in both precision and recall to those produced by the BWA, MarkDuplicates, BQSR, and HaplotypeCaller commands in the GATK's Whole Genome Germline Single Sample variant calling workflow.

Given one or more pairs of FASTQ files, you can run the germline variant tool to generate BAM, variants, duplicate metrics and recal.

The germline pipeline shown below resembles the GATK4 best practices pipeline. The inputs are BWA-indexed reference files, pair-ended FASTQ files, and knownSites for BQSR calculation. The outputs of this pipeline are as follows:

- Aligned, co-ordinate sorted, duplicated marked BAM
- BQSR report
- Variants in `vcf` / `g.vcf` / `g.vcf.gz` format



Quick Start

Running the germline pipeline:

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun germline \ --ref /workdir/${REFERENCE_FILE} \ --in-fq
/workdir/${INPUT_FASTQ_1} /workdir/${INPUT_FASTQ_2} \ --knownSites
/workdir/${KNOWN_SITES_FILE} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-
variants /outputdir/${OUTPUT_VCF} \ --out-recal-file /outputdir/${OUT_RECAL_FILE}
```

Specifying Haplotype Caller options

Several original HaplotypeCaller options are supported by Parabricks. To specify the inclusion or exclusion of several haplotype caller annotations, use the `--haplotypecaller-options` option:

```
$ # 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun haplotypecaller \ ... --haplotypecaller-options '-min-
pruning 4 -A AS_BaseQualityRankSumTest -A TandemRepeat' ...
```

Annotations may be excluded in the same manner using the `-AX` option. There should be a space between the `-A`/`-AX` flag and its value.

The following are supported options and their allowed values:

- -A
  - AS_BaseQualityRankSumTest
  - AS_FisherStrand
  - AS_InbreedingCoeff
  - AS_MappingQualityRankSumTest
  - AS_QualByDepth
  - AS_RMSMappingQuality
  - AS_ReadPosRankSumTest
  - AS_StrandOddsRatio
  - BaseQualityRankSumTest
  - ChromosomeCounts
  - ClippingRankSumTest
  - Coverage
  - DepthPerAlleleBySample
  - DepthPerSampleHC
  - ExcessHet
  - FisherStrand
  - InbreedingCoeff
  - MappingQualityRankSumTest
  - QualByDepth
  - RMSMappingQuality
  - ReadPosRankSumTest

    - ReferenceBases
    - StrandBiasBySample
    - StrandOddsRatio
    - TandemRepeat
- -A*X*
    - (same as for the *-A* option)
- --output-mode
    - EMIT_VARIANTS_ONLY
    - EMIT_ALL_CONFIDENT_SITES
    - EMIT_ALL_ACTIVE_SITES
- -m*ax-reads-per-alignment-start*
    - a positive integer
- -m*in-dangling-branch-length*
    - a positive integer
- -m*in-pruning*
    - a positive integer
- -p*cr-indel-model*
    - NONE
    - HOSTILE
    - AGGRESSIVE
    - CONSERVATIVE

- *-standard-min-confidence-threshold-for-calling*
    - a positive integer

Compatible CPU-based BWA-MEM, GATK4 Commands

The commands below are the bwa-0.7.12 and GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# Run bwa-mem and pipe output to create sorted BAM $ bwa mem \ -t 32 \ -K
10000000 \ -R '@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \
<INPUT_DIR>/${REFERENCE_FILE} <INPUT_DIR>/${INPUT_FASTQ_1}
<INPUT_DIR>/${INPUT_FASTQ_2} | \ gatk SortSam \ --java-options -Xmx30g \ --
MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --SORT_ORDER
coordinate # Mark Duplicates $ gatk MarkDuplicates \ --java-options -Xmx30g \ -I
cpu.bam \ -O mark_dups_cpu.bam \ -M metrics.txt # Generate BQSR Report $ gatk
BaseRecalibrator \ --java-options -Xmx30g \ --input mark_dups_cpu.bam \ --output
<OUTPUT_DIR>/${OUT_RECAL_FILE} \ --known-sites
<INPUT_DIR>/${KNOWN_SITES_FILE} \ --reference <INPUT_DIR>/${REFERENCE_FILE}
# Run ApplyBQSR Step $ gatk ApplyBQSR \ --java-options -Xmx30g \ -R
<INPUT_DIR>/${REFERENCE_FILE} \ -I mark_dups_cpu.bam \ --bqsr-recal-file
<OUTPUT_DIR>/${OUT_RECAL_FILE} \ -O cpu_nodups_BQSR.bam #Run Haplotype
Caller $ gatk HaplotypeCaller \ --java-options -Xmx30g \ --input
cpu_nodups_BQSR.bam \ --output <OUTPUT_DIR>/${OUTPUT_VCF} \ --reference
<INPUT_DIR>/${REFERENCE_FILE} \ --native-pair-hmm-threads 16
```

germline Reference

Run Germline pipeline to convert FASTQ to VCF.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-fq [IN_FQ ...]

Path to the pair-ended FASTQ files followed by optional read groups with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The files must be in fastq or fastq.gz format. All sets of inputs should have a read group; otherwise, none should have a read group, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-fq [IN_SE_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-se-fq sampleX_1.fastq.gz --in-se-fq sampleX_2.fastq.gz . Example 2: --in-se-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-se-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of the report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of BAM file after Marking Duplicates. (default: None)

Option is required.

--htvc-bam-output HTVC_BAM_OUTPUT

File to which assembled haplotypes should be written in HaplotypeCaller. (default: None)

--out-variants OUT_VARIANTS

Path of the vcf/gvcf/gvcf.gz file after variant calling. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of duplicate metrics file after Marking Duplicates. (default: None)

Tool Options:

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000". (default: None)

--bwa-options BWA_OPTIONS

Pass supported bwa mem options as one string. The current original bwa mem supported options are -M, -Y and -T e.g. --bwa-options="-M -Y" (default: None)

--no-warnings

Suppress warning messages about system thread and memory usage. (default: None)

--filter-flag FILTER_FLAG

Don't generate SAM entries in the output if the entry's flag's meet this criteria. Criteria: (flag & filter != 0) (default: 0)

--skip-multiple-hits

Filter SAM entries whose length of SA is not 0. (default: None)

--min-read-length MIN_READ_LENGTH

Skip reads below minimum read length. They will not be part of the output. (default: None)

--align-only

Generate output BAM after bwa-mem. The output will not be co-ordinate sorted or duplicates will not be marked. (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--fix-mate

Add mate cigar (MC) and mate quality (MQ) tags to the output file. (default: None)

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--markdups-picard-version-2182

Assume marking duplicates to be similar to Picard version 2.18.2. (default: None)

--monitor-usage

Monitor approximate CPU utilization and host memory usage during execution. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. Ignored if --out-duplicate-metrics is not passed. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR. (default: None)

--max-read-length-fq2bamfast MAX_READ_LENGTH_FQ2BAMFAST

Maximum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (Argument only applies to --fq2bamfast) (default: 480)

--min-read-length-fq2bamfast MIN_READ_LENGTH_FQ2BAMFAST

Minimum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (Argument only applies to --fq2bamfast) (default: 10)

--haplotypecaller-options HAPLOTYPECALLER_OPTIONS

Pass supported haplotype caller options as one string. The following are currently supported original haplotypecaller options: -A <AS_BaseQualityRankSumTest, AS_FisherStrand, AS_InbreedingCoeff, AS_MappingQualityRankSumTest, AS_QualByDepth, AS_RMSMappingQuality, AS_ReadPosRankSumTest, AS_StrandOddsRatio, BaseQualityRankSumTest, ChromosomeCounts, ClippingRankSumTest, Coverage, DepthPerAlleleBySample, DepthPerSampleHC, ExcessHet, FisherStrand, InbreedingCoeff, MappingQualityRankSumTest, QualByDepth, RMSMappingQuality, ReadPosRankSumTest, ReferenceBases, StrandBiasBySample, StrandOddsRatio, TandemRepeat>,-AX <same options as -A>,--output-mode <EMIT_VARIANTS_ONLY, EMIT_ALL_CONFIDENT_SITES, EMIT_ALL_ACTIVE_SITES> ,-max-reads-per-alignment-start <int>, -min-dangling-branch-length <int>, -min-pruning <int>, -pcr-indel-model <NONE, HOSTILE, AGGRESSIVE, CONSERVATIVE>, -standard-min-confidence-threshold-for-calling <int>(e.g. --haplotypecaller-options="-min-pruning 4 -standard-min-confidence-threshold-for-calling 30"). (default: None)

--static-quantized-quals STATIC_QUANTIZED_QUALS

Use static quantized quality scores to a given number of levels. Repeat this option multiple times for multiple bins. (default: None)

--gvcf

Generate variant calls in .gvcf Format. (default: None)

--disable-read-filter DISABLE_READ_FILTER

Disable the read filters for BAM entries. Currently, the supported read filters that can be disabled are MappingQualityAvailableReadFilter, MappingQualityReadFilter, NotSecondaryAlignmentReadFilter, and WellformedReadFilter. (default: None)

--max-alternate-alleles MAX_ALTERNATE_ALLELES

Maximum number of alternate alleles to genotype. (default: None)

-G ANNOTATION_GROUP, --annotation-group ANNOTATION_GROUP

The groups of annotations to add to the output variant calls. Currently supported annotation groups are StandardAnnotation, StandardHCAnnotation, and AS_StandardAnnotation. (default: None)

-GQB GVCF_GQ_BANDS, --gvcf-gq-bands GVCF_GQ_BANDS

Exclusive upper bounds for reference confidence GQ bands. Must be in the range [1, 100] and specified in increasing order. (default: None)

--rna

Run haplotypecaller optimized for RNA data. (default: None)

--dont-use-soft-clipped-bases

Don't use soft clipped bases for variant calling. (default: None)

--minimum-mapping-quality MINIMUM_MAPPING_QUALITY

Minimum mapping quality to keep (inclusive). (default: None)

--mapping-quality-threshold-for-genotyping MAPPING_QUALITY_THRESHOLD_FOR_GENOTYPING

Control the threshold for discounting reads from the genotyper due to mapping quality after the active region detection and assembly steps but before genotyping. (default: None)

--enable-dynamic-read-disqualification-for-genotyping

Will enable less strict read disqualification low base quality reads. (default: None)

--no-alt-contigs

Get rid of output records for alternate contigs. (default: None)

--ploidy PLOIDY

Ploidy assumed for the BAM file. Currently only haploid (ploidy 1) and diploid (ploidy 2) are supported. (default: 2)

--sample-sex SAMPLE_SEX

Sex of the sample input. This option will override the sex determined from any X/Y read ratio range. Must be either male or female. (default: None)

--range-male RANGE_MALE

Inclusive male range for the X/Y read ratio. The sex is declared male if the actual ratio falls in the specified range. Syntax is "<min>-<max>" (e.g. "--range-male 1-10"). (default: None)

--range-female RANGE_FEMALE

Inclusive female range for the X/Y read ratio. The sex is declared female if the actual ratio falls in the specified range. Syntax is "<min>-<max>" (e.g. "--range-female 150-250"). (default: None)

--use-GRCh37-regions

Use the pseudoautosomal regions for GRCh37 reference types. This flag should be used for GRCh37 and UCSC hg19 references. By default, GRCh38 regions are used.

(default: None)

Performance Options:

--fq2bamfast

Use fq2bamfast as the alignment tool instead of fq2bam (default: None)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting and marking. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--memory-limit MEMORY_LIMIT

System memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

--low-memory

Use low memory mode (default: None)

--num-cpu-threads-per-stage NUM_CPU_THREADS_PER_STAGE

Number of CPU threads to use per stage. (default: 8)

--bwa-nstreams BWA_NSTREAMS

Number of streams per GPU to use; note: more streams increases device memory usage (Argument only applies to --fq2bamfast) (default: 4)

--bwa-cpu-thread-pool BWA_CPU_THREAD_POOL

Number of threads to devote to CPU thread pool *per GPU* (Argument only applies to --fq2bamfast) (default: 16)

--htvc-low-memory

Use low memory mode in htvc. (default: None)

--num-htvc-threads NUM_HTVC_THREADS

Number of CPU threads. (default: 5)

--run-partition

Divide the whole genome into multiple partitions and run multiple processes at the same time, each on one partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--read-from-tmp-dir

Running variant caller reading from bin files generated by Aligner and sort. Run postsort in parallel. This option will increase device memory usage. (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

ⓘ **Note**

The *--in-fq* option takes the names of two FASTQ files, optionally followed by a quoted read group. The FASTQ filenames must not start with a hyphen.

ⓘ **Note**

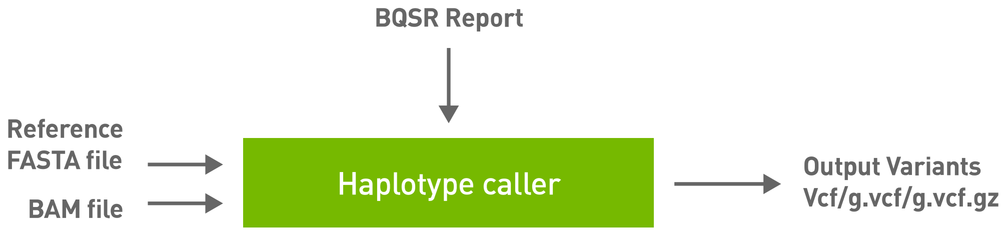
In the values provided to *--haplotypecaller-options* --output-mode requires two leading hyphens, while all other values take a single hyphen.

haplotypecaller

Run a GPU-accelerated haplotypecaller.

This tool applies an accelerated GATK CollectMultipleMetrics for assessing the metrics of a BAM file, such as including alignment success, quality score distributions, GC bias, and sequencing artifacts. This functions as a ‘meta-metrics’ tool, and can run any combination of the available metrics tools in GATK to assess overall how well a sequencing run has performed. The available metrics tools (PROGRAMs) can be found in the command line example below.

You can provide an optional BQSR report to fix the BAM, similar to ApplyBQSR. In this case, the updated base qualities will be used.



Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun haplotypecaller \ --ref /workdir/${REFERENCE_FILE} \ --in-bam /workdir/${INPUT_BAM} \ --in-recal-file /workdir/${INPUT_RECAL_FILE} \ --out-variants /outputdir/${OUTPUT_VCF}
```

Compatible GATK4 Command

The commands below are the GATK4 counterpart of the Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# Run ApplyBQSR Step $ gatk ApplyBQSR \ --java-options -Xmx30g \ -R Ref/Homo_sapiens_assembly38.fasta \ -I mark_dups_cpu.bam \ --bqsr-recal-file recal_file.txt \ -O cpu_nodups_BQSR.bam #Run Haplotype Caller $ gatk HaplotypeCaller \ --java-options -Xmx30g \ --input cpu_nodups_BQSR.bam \ --output result_cpu.vcf \ --reference Ref/Homo_sapiens_assembly38.fasta \ --native-pair-hmm-threads 16
```

Specifying Haplotype Caller options

Several original HaplotypeCaller options are supported by Parabricks. To specify the inclusion or exclusion of several haplotype caller annotations, use the `--haplotypecaller-options` option:

```
$ # 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun haplotypecaller \ ... --haplotypecaller-options '-min-pruning 4 -A AS_BaseQualityRankSumTest -A TandemRepeat' ...
```

Annotations may be excluded in the same manner using the `-AX` option. There should be a space between the `-A`/`-AX` flag and its value.

The following are supported options and their allowed values:

- -A
  - AS_BaseQualityRankSumTest
  - AS_FisherStrand
  - AS_InbreedingCoeff
  - AS_MappingQualityRankSumTest
  - AS_QualByDepth
  - AS_RMSMappingQuality
  - AS_ReadPosRankSumTest
  - AS_StrandOddsRatio
  - BaseQualityRankSumTest
  - ChromosomeCounts
  - ClippingRankSumTest
  - Coverage
  - DepthPerAlleleBySample
  - DepthPerSampleHC
  - ExcessHet
  - FisherStrand
  - InbreedingCoeff
  - MappingQualityRankSumTest
  - QualByDepth
  - RMSMappingQuality
  - ReadPosRankSumTest

    - ReferenceBases
    - StrandBiasBySample
    - StrandOddsRatio
    - TandemRepeat
- -A*X*
    - (same as for the *-A* option)
- --output-mode
    - EMIT_VARIANTS_ONLY
    - EMIT_ALL_CONFIDENT_SITES
    - EMIT_ALL_ACTIVE_SITES
- -m*ax-reads-per-alignment-start*
    - a positive integer
- -m*in-dangling-branch-length*
    - a positive integer
- -m*in-pruning*
    - a positive integer
- -p*cr-indel-model*
    - NONE
    - HOSTILE
    - AGGRESSIVE
    - CONSERVATIVE

- *-standard-min-confidence-threshold-for-calling*
    - a positive integer

haplotypecaller Reference

Run HaplotypeCaller to convert BAM/CRAM to VCF.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-bam IN_BAM

Path to the input BAM/CRAM file for variant calling. The argument may also be a local folder containing several BAM files. (default: None)

Option is required.

--in-recal-file IN_RECAL_FILE

Path to the input BQSR report. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--htvc-bam-output HTVC_BAM_OUTPUT

File to which assembled haplotypes should be written. (default: None)

--out-variants OUT_VARIANTS

Path of the vcf/g.vcf/gvcf.gz file after variant calling. (default: None)

Option is required.

Tool Options:

--haplotypecaller-options HAPLOTYPECALLER_OPTIONS

Pass supported haplotype caller options as one string. The following are currently supported original haplotypecaller options: -A <AS_BaseQualityRankSumTest, AS_FisherStrand, AS_InbreedingCoeff, AS_MappingQualityRankSumTest, AS_QualByDepth, AS_RMSMappingQuality, AS_ReadPosRankSumTest, AS_StrandOddsRatio, BaseQualityRankSumTest, ChromosomeCounts, ClippingRankSumTest, Coverage, DepthPerAlleleBySample, DepthPerSampleHC, ExcessHet, FisherStrand, InbreedingCoeff, MappingQualityRankSumTest, QualByDepth, RMSMappingQuality, ReadPosRankSumTest, ReferenceBases, StrandBiasBySample, StrandOddsRatio, TandemRepeat>,-AX <same options as -A>,--output-mode <EMIT_VARIANTS_ONLY, EMIT_ALL_CONFIDENT_SITES, EMIT_ALL_ACTIVE_SITES> ,-max-reads-per-alignment-start <int>, -min-dangling-branch-length <int>, -min-pruning <int>, -pcr-indel-model <NONE, HOSTILE, AGGRESSIVE, CONSERVATIVE>, -standard-min-confidence-threshold-for-calling <int>(e.g. --haplotypecaller-options="-min-pruning 4 -standard-min-confidence-threshold-for-calling 30"). (default: None)

--static-quantized-quals STATIC_QUANTIZED_QUALS

Use static quantized quality scores to a given number of levels. Repeat this option multiple times for multiple bins. (default: None)

--gvcf

Generate variant calls in .gvcf Format. (default: None)

--disable-read-filter DISABLE_READ_FILTER

Disable the read filters for BAM entries. Currently, the supported read filters that can be disabled are MappingQualityAvailableReadFilter, MappingQualityReadFilter, NotSecondaryAlignmentReadFilter, and WellformedReadFilter. (default: None)

--max-alternate-alleles MAX_ALTERNATE_ALLELES

Maximum number of alternate alleles to genotype. (default: None)

-G ANNOTATION_GROUP, --annotation-group ANNOTATION_GROUP

The groups of annotations to add to the output variant calls. Currently supported annotation groups are StandardAnnotation, StandardHCAnnotation, and AS_StandardAnnotation. (default: None)

-GQB GVCF_GQ_BANDS, --gvcf-gq-bands GVCF_GQ_BANDS

Exclusive upper bounds for reference confidence GQ bands. Must be in the range [1, 100] and specified in increasing order. (default: None)

--rna

Run haplotypecaller optimized for RNA data. (default: None)

--dont-use-soft-clipped-bases

Don't use soft clipped bases for variant calling. (default: None)

--minimum-mapping-quality MINIMUM_MAPPING_QUALITY

Minimum mapping quality to keep (inclusive). (default: None)

--mapping-quality-threshold-for-genotyping MAPPING_QUALITY_THRESHOLD_FOR_GENOTYPING

Control the threshold for discounting reads from the genotyper due to mapping quality after the active region detection and assembly steps but before genotyping. (default: None)

--enable-dynamic-read-disqualification-for-genotyping

Will enable less strict read disqualification low base quality reads. (default: None)

--no-alt-contigs

Get rid of output records for alternate contigs. (default: None)

--ploidy PLOIDY

Ploidy assumed for the BAM file. Currently only haploid (ploidy 1) and diploid (ploidy 2) are supported. (default: 2)

-L INTERVAL, --interval INTERVAL

Interval within which to call the variants from the BAM/CRAM file. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--sample-sex SAMPLE_SEX

Sex of the sample input. This option will override the sex determined from any X/Y read ratio range. Must be either male or female. (default: None)

--range-male RANGE_MALE

Inclusive male range for the X/Y read ratio. The sex is declared male if the actual ratio falls in the specified range. Syntax is "<min>-<max>" (e.g. "--range-male 1-10"). (default: None)

--range-female RANGE_FEMALE

Inclusive female range for the X/Y read ratio. The sex is declared female if the actual ratio falls in the specified range. Syntax is "<min>-<max>" (e.g. "--range-female 150-250"). (default: None)

--use-GRCh37-regions

Use the pseudoautosomal regions for GRCh37 reference types. This flag should be used for GRCh37 and UCSC hg19 references. By default, GRCh38 regions are used.

(default: None)

Performance Options:

--htvc-low-memory

Use low memory mode in htvc. (default: None)

--num-htvc-threads NUM_HTVC_THREADS

Number of CPU threads. (default: 5)

--run-partition

Divide the whole genome into multiple partitions and run multiple processes at the same time, each on one partition. (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

Note

In the values provided to *--haplotypecaller-options* --output-mode requires two leading hyphens, while all other values take a single hyphen.

indexgvcf

This tool creates an index for `g.vcf`/`g.vcf.gz` files. The index file name is determined by appending `.tbi` to the name of the GVCF file being indexed, and is created in the same directory as the index file itself.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun indexgvcf \ --input /workdir/${INPUT_GVCF}
```

Compatible CPU GATK4 Command

```
$ gatk IndexFeatureFile -I <INPUT_DIR>/${INPUT_GVCF}
```

indexgvcf Reference

Index a GVCF file.

Input/Output file options

--input INPUT

Path to the g.vcf/g.vcf.gz file to be indexed.

(default: None)

Option is required.

Options specific to this tool

(none)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

markdup

Mark duplicated reads in a BAM/CRAM file.

This tool locates and tags duplicate reads in a BAM or SAM file, where duplicate reads are defined as originating from a single fragment of DNA.

markdup supports the marking of duplicates in two ways, assuming the sort order to be coordinate (the default) or queryname (**--markdups-assume-sortorder-querynamer**).

The input BAM/CRAM must be sorted by queryname. If it is not, please run **pbrun bamsort** with **--sort-order queryname** to preprocess the input file.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun markdup \ --ref /workdir/${REFERENCE_FILE} \ --in-bam
/workdir/${INPUT_BAM} \ --out-bam /outputdir/${OUTPUT_BAM}
```

Compatible Baseline Command

The command below is the GATK counterpart of the Parabricks command above. Note that the corresponding baseline command is different between marking by coordinate and by queryname. Choose the correct one based on your case. The first **gatk SortSam** command is listed here to guarantee the order of the input file to MarkDuplicates. Feel free to ignore it if your file order is correct.

Coordinate Sort Order

```
gatk SortSam \ -R <INPUT_DIR>/${REFERENCE_FILE} \ -I <INPUT_DIR>/${INPUT_BAM}
\ -O <INPUT_DIR>/${SORTED_BAM} \ -SO coordinate gatk MarkDuplicates \ -I
<INPUT_DIR>/${SORTED_BAM} \ -O <OUTPUT_DIR>/${MARKED_BAM} \ -M
<OUTPUT_DIR>/${METRICS_FILE} \ -ASO coordinate
```

Queryname Sort Order

```
gatk SortSam \ -R <INPUT_DIR>/${REFERENCE_FILE} \ -I <INPUT_DIR>/${INPUT_BAM}
\ -O <INPUT_DIR>/${SORTED_BAM} \ -SO queryname gatk MarkDuplicates \ -I
<INPUT_DIR>/${SORTED_BAM} \ -O <OUTPUT_DIR>/${MARKED_BAM} \ -M
<OUTPUT_DIR>/${METRICS_FILE} \ -ASO queryname gatk SortSam \ -R
<INPUT_DIR>/${REFERENCE_FILE} \ -I <OUTPUT_DIR>/${MARKED_BAM} \ -O
<OUTPUT_DIR>/${FINAL_BAM} \ -SO coordinate
```

markdup Reference

Mark duplicate reads in BAM file. The input file should be sorted by queryname.

Input/Output file options

--in-bam IN_BAM

Path of BAM/CRAM for marking duplicate. Need to be sorted by queryname already. This option is required. (default: None)

Option is required.

--out-bam OUT_BAM

Path of BAM/CRAM file after marking duplicate. (default: None)

Option is required.

--ref REF

Path to the reference file. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of duplicate metrics file after Marking Duplicates. (default: None)

Tool Options:

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. (default: None)

Performance Options:

--num-zip-threads NUM_ZIP_THREADS

Number of CPUs to use for zipping BAM/CRAM files in a run (default 10). (default: None)

--num-worker-threads NUM_WORKER_THREADS

Number of CPUs to use for markdup in a run (default 10). (default: None)

--mem-limit MEM_LIMIT

Memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

minimap2 (Beta)

Run a GPU-accelerated minimap2.

This tool aligns long read sequences against a large reference database using an accelerated KSW2 to convert FASTQ to BAM/CRAM.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun minimap2 \ --ref /workdir/${REFERENCE_FILE} \ --in-fq
/workdir/${INPUT_FASTQ} \ --out-bam /outputdir/${OUTPUT_BAM}
```

Compatible CPU-based minimap2, GATK4 Commands

The commands below are the minimap2-v2.26 and GATK4 counterpart of the Clara Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results. You may need to increase the Java heap size based on your dataset, or decrease the number of --MAX_RECORDS_IN_RAM.

```
# Run minimap2 and pipe the output to create a sorted BAM. $ minimap2 -ax map-pbmm2 \ <INPUT_DIR>/${REFERENCE_FILE} \ <INPUT_DIR>/${INPUT_FASTQ} | \ gatk SortSam \ --java-options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --SORT_ORDER coordinate
```

Please note that two changes must be made to the baseline minimap2 code in order to match the results exactly:

Firstly, a new preset must be made in `options.c` in the `mm_set_opt` function that tries to replicate the preset of pbmm2 by setting these parameters as a new preset named "map-pbmm2":

```
io->k = 19; io->w = 10; io->batch_size = 0x7fffffffffffffffL; // always build a uni-part index mo->flag |= MM_I_HPC; mo->flag |= MM_F_CIGAR; mo->flag |= MM_F_LONG_CIGAR; mo->flag |= MM_F_EQX; mo->flag |= MM_F_SOFTCLIP; mo->flag |= MM_F_NO_PRINT_2ND; mo->flag |= MM_F_HARD_MLEVEL; mo->mask_level = 0; mo->e2 = 1; mo->zdrop = 400; mo->a = 2; mo->b = 5; mo->q = 5; mo->q2 = 56; mo->e = 4; mo->zdrop_inv = 50; mo->bw = 2000;
```

Secondly, a fix must be made to the baseline KSW2 code to round the loop fission start and end points by changing them to `st` and `en` respectively. If the start point (`st0`) is a number below 16, but greater than 0, its scoring values will not be initialized correctly, but will still be used later when computing the actual alignment. This can be fixed by rounding the start and end points to multiples of 16.

To make this fix, change the following code in `ksw2_extd2_sse.c`:

```
// loop fission: set scores first if (!(flag & KSW_EZ_GENERIC_SC)) { for (t = st0; t <= en0;
t += 16) { __m128i sq, st, tmp, mask; sq = _mm_loadu_si128((__m128i*)&sf[t]); st =
_mm_loadu_si128((__m128i*)&qrr[t]); mask = _mm_or_si128(_mm_cmpeq_epi8(sq,
m1_), _mm_cmpeq_epi8(st, m1_)); tmp = _mm_cmpeq_epi8(sq, st); #ifdef __SSE4_1__
tmp = _mm_blendv_epi8(sc_mis_, sc_mch_, tmp); tmp = _mm_blendv_epi8(tmp,
sc_N_, mask); #else tmp = _mm_or_si128(_mm_andnot_si128(tmp, sc_mis_),
_mm_and_si128(tmp, sc_mch_)); tmp = _mm_or_si128(_mm_andnot_si128(mask,
tmp), _mm_and_si128(mask, sc_N_)); #endif _mm_storeu_si128((__m128i*)((int8_t*)s
+ t), tmp); } } else { for (t = st0; t <= en0; ++t) ((uint8_t*)s)[t] = mat[sf[t] * m + qrr[t]]; }
```

Fixed version that uses `lf_start` and `lf_en`:

```
// loop fission: set scores first int lf_start = st, lf_en = en; if (!(flag &
KSW_EZ_GENERIC_SC)) { for (t = lf_start; t <= lf_en; t += 16) { __m128i sq, st, tmp,
mask; sq = _mm_loadu_si128((__m128i*)&sf[t]); st =
_mm_loadu_si128((__m128i*)&qrr[t]); mask = _mm_or_si128(_mm_cmpeq_epi8(sq,
m1_), _mm_cmpeq_epi8(st, m1_)); tmp = _mm_cmpeq_epi8(sq, st); #ifdef __SSE4_1__
tmp = _mm_blendv_epi8(sc_mis_, sc_mch_, tmp); tmp = _mm_blendv_epi8(tmp,
sc_N_, mask); #else tmp = _mm_or_si128(_mm_andnot_si128(tmp, sc_mis_),
_mm_and_si128(tmp, sc_mch_)); tmp = _mm_or_si128(_mm_andnot_si128(mask,
tmp), _mm_and_si128(mask, sc_N_)); #endif _mm_storeu_si128((__m128i*)((int8_t*)s
+ t), tmp); } } else { for (t = lf_start; t <= lf_en; ++t) ((uint8_t*)s)[t] = mat[sf[t] * m +
qrr[t]]; }
```

minimap2 Reference

Align long read sequences against a large reference database to convert FASTQ to BAM/CRAM.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--index INDEX

Path to a minimizer index file generated by vanilla minimap2 to reduce indexing time. (default: None)

--in-fq IN_FQ

Path to a query sequence file in fastq or fastq.gz format. (default: None)

Option is required.

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of a report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of a BAM/CRAM file after sorting. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of duplicate metrics file after Marking Duplicates. (default: None)

--out-qc-metrics-dir OUT_QC_METRICS_DIR

Path of the directory where QC metrics will be generated. (default: None)

Tool Options:

--preset PRESET

Which preset to apply. Possible values are {map-pbmm2,map-hifi,map-ont}. (default: map-pbmm2)

--eqx

Write =/X CIGAR operators. (default: None)

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR after generating sorted BAM. This option requires both --knownSites and --out-recal-file input parameters. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

Performance Options:

--num-threads NUM_THREADS

Number of processing threads. (default: 128)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--low-memory

Use low memory mode (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

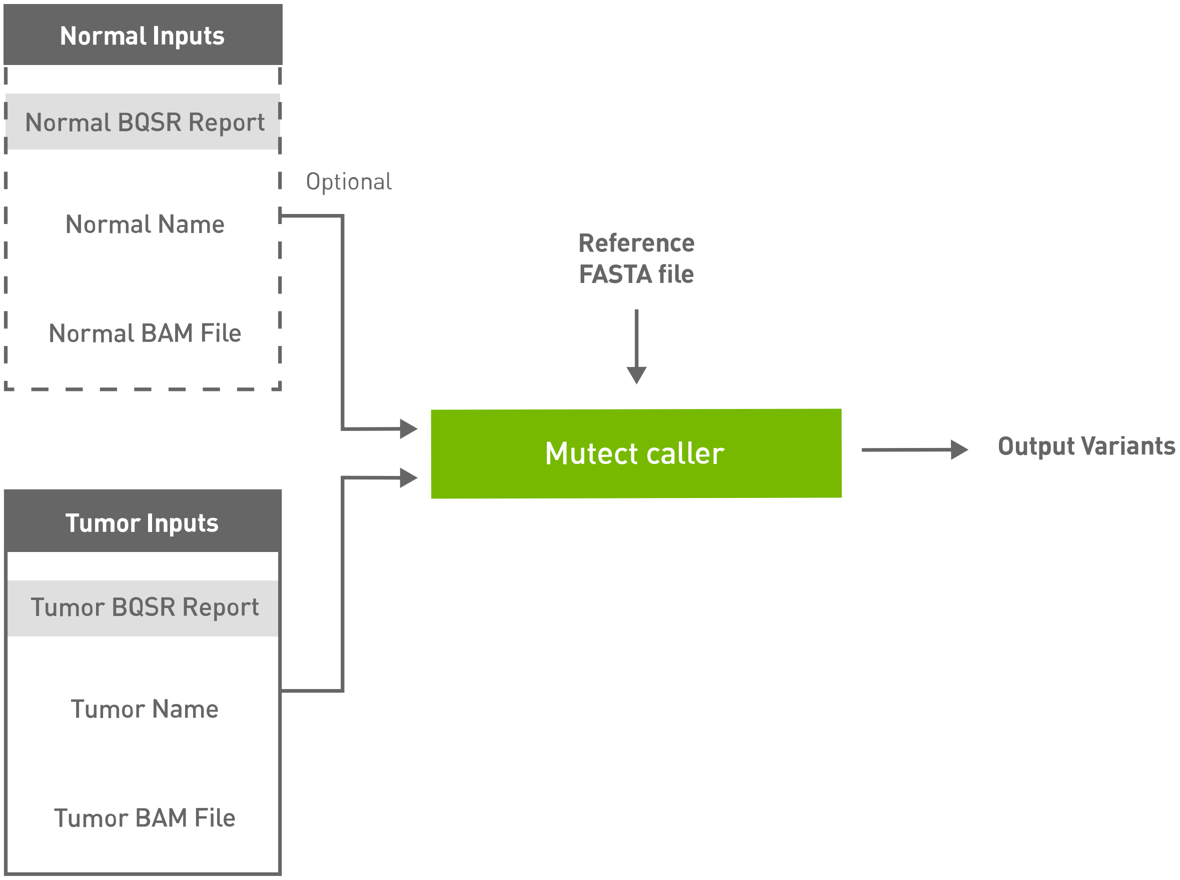
--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

mutectcaller

This tool is an accelerated version of the GATK somatic variant caller, Mutect2, which takes aligned BAMs from the FQ2BAM tool, and outputs a VCF file. This can take as input either a single (“tumor-only”) BAM, or a pair of BAMs (“tumor-normal”) to provide a baseline to call somatic variants against.

The figure below shows the high-level functionality of mutectcaller. All dotted boxes indicate optional data, with some constraints.



The names of the tumor sample (for the `--tumor-name` option) and the normal sample (for the `--normal-name` option) can be extracted from the headers of their respective BAM files with samtools, which can be installed through apt-get:

```
$ sudo apt-get install samtools
```

Or you can build it from source codes by following the instructions in samtools repo.

Once you have samtools installed on your system you can run this command to get the sample name (SM) field:

```
$ samtools view NA12878.bam -H | grep '@RG' @RG ID:HJYFJ.4 SM:NA12878
LB:Pond-492093 PL:illumina PU:HJYFJCCXX160204.4.GCCGCAAC CN:BI DT:2016-02-
```

```
04T00:00:00-0500
```

The sample name is the value after "*SM:*" (NA12878, in this example)

If there are multiple read group (@RG) lines and all of them have the same sample name you may safely use the common sample name. If there are multiple read group lines with multiple sample names, choose one sample name as the input. All reads with that sample name will be processed by `mutectcaller` and all other reads will be ignored. Currently only one sample name per BAM file is supported.

If there are no read group lines in the BAM header, or there is no sample name in the read group line, you will need to add read group information to the BAM file. This may be done by running this command:

```
$ samtools addreplacerg \ -r
"@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample_sm\tPU:sample_rg1" \
original_file.bam \ -o updated_file.bam \ -O BAM
```

This will update the sample name of all reads in this BAM file to "sample_sm", and you can pass "sample_sm" as the sample name of this BAM file. Make sure you use the *updated_file.bam* as input to `mutectcaller`.

Quick Start

You can download the mutect sample dataset from here. Extract all files by running:

```
$ tar -xvzf mutect_sample.tar.gz mutect_sample/
mutect_sample/germline_resource.vcf.gz.tbi mutect_sample/force_call.vcf.gz.tbi
mutect_sample/germline_resource.vcf.gz mutect_sample/tumor.bam.bai
mutect_sample/GCA_000001405.15_GRCh38_no_alt_analysis_set.fa
mutect_sample/force_call.vcf.gz mutect_sample/tumor.bam
mutect_sample/normal.bam.bai mutect_sample/normal.bam
```

Inside the `mutect_sample` folder you will find the necessary input files including:

- one reference fasta (GCA_000001405.15_GRCh38_no_alt_analysis_set.fa),
- one tumor bam (tumor.bam),
- one normal bam (normal.bam),
- one force_calling.vcf.gz VCF file and
- one germline_resource.vcf.gz VCF file

with all necessary indexes.

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun mutectcaller \ --ref /workdir/${REFERENCE_FILE} \ --
tumor-name tumor_name_inside_bam_file \ --in-tumor-bam
/workdir/${INPUT_TUMOR_BAM} \ --in-normal-bam
/workdir/${INPUT_NORMAL_BAM} \ --normal-name normal_name_inside_bam_file \
--out-vcf /outputdir/${OUTPUT_VCF}
```

Compatible GATK4 Command

The command below is the GATK4 counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
$ gatk Mutect2 \ -R <INPUT_DIR>/${REFERENCE_FILE} \ --input
<INPUT_DIR>/${INPUT_TUMOR_BAM} \ --tumor-sample
tumor_name_inside_bam_file \ --input <INPUT_DIR>/${INPUT_NORMAL_BAM} \ --
normal-sample normal_name_inside_bam_file \ --output
<OUTPUT_DIR>/${OUTPUT_VCF}
```

Mutect2 with Panel of Normals

Parabricks Mutect2 from version 3.7.0-1 has started supporting Panel of Normals to process variants. There are three steps involved:

- prepon
- running mutectcaller with the index generated by prepon
- postpon, updating the vcf with pon information

```
# The first command will generate input.pon that should be done once for the
input.vcf.gz # 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun prepon --in-pon-file /workdir/${INPUT_PON_VCF} # Run
mutectcaller with the pon index # This command assumes all the inputs are in
INPUT_DIR and all the outputs go to OUTPUT_DIR. docker run --rm --gpus all --volume
INPUT_DIR:/workdir --volume OUTPUT_DIR:/outputdir \ --workdir /workdir \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun mutectcaller \ --ref
/workdir/${REFERENCE_FILE} \ --tumor-name tumor \ --in-tumor-bam
/workdir/${INPUT_TUMOR_BAM} \ --in-normal-bam
/workdir/${INPUT_NORMAL_BAM} \ --pon /workdir/${INPUT_PON_VCF} \ --normal-name normal \ --out-vcf /outputdir/${OUTPUT_VCF} # Add the annotation to the
output.vcf generated above # 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 \ --workdir /workdir \
nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun postpon \ --in-vcf
/workdir/${OUTPUT_VCF} \ --in-pon-file /workdir/${INPUT_PON_FILE} \ --out-vcf
/outputdir/${OUTPUT_ANNOTATED_VCF}
```

mutectcaller Reference

Run GPU mutect2 to convert BAM/CRAM to vcf

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--out-vcf OUT_VCF

Path of the VCF file after Variant Calling. (default: None)

Option is required.

--in-tumor-bam IN_TUMOR_BAM

Path of the BAM/CRAM file for tumor reads. (default: None)

Option is required.

--in-normal-bam IN_NORMAL_BAM

Path of the BAM/CRAM file for normal reads. (default: None)

--in-tumor-recal-file IN_TUMOR_RECAL_FILE

Path of the report file after Base Quality Score Recalibration for tumor sample. (default: None)

--in-normal-recal-file IN_NORMAL_RECAL_FILE

Path of the report file after Base Quality Score Recalibration for normal sample. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--mutect-bam-output MUTECT_BAM_OUTPUT

File to which assembled haplotypes should be written. (default: None)

--pon PON

Path of the vcf.gz PON file. Make sure you run prepon first and there is a '.pon' file already. (default: None)

--mutect-germline-resource MUTECT_GERMLINE_RESOURCE

Path of the vcf.gz germline resource file. Population vcf of germline sequencing containing allele fractions. (default: None)

--mutect-alleles MUTECT_ALLELES

Path of the vcf.gz force-call file. The set of alleles to force-call regardless of evidence. (default: None)

Tool Options:

--max-mnp-distance MAX_MNP_DISTANCE

Two or more phased substitutions separated by this distance or less are merged into MNPs. (default: 1)

--mutectcaller-options MUTECTCALLER_OPTIONS

Pass supported mutectcaller options as one string. The following are currently supported original mutectcaller options: -pcr-indel-model <NONE, HOSTILE, AGGRESSIVE, CONSERVATIVE>, -max-reads-per-alignment-start <int>, (e.g. --mutectcaller-options="-pcr-indel-model HOSTILE -max-reads-per-alignment-start 30"). (default: None)

--initial-tumor-lod INITIAL_TUMOR_LOD

Log 10 odds threshold to consider pileup active. (default: None)

--tumor-lod-to-emit TUMOR_LOD_TO_EMIT

Log 10 odds threshold to emit variant to VCF. (default: None)

--pruning-lod-threshold PRUNING_LOD_THRESHOLD

Ln likelihood ratio threshold for adaptive pruning algorithm. (default: None)

--active-probability-threshold ACTIVE_PROBABILITY_THRESHOLD

Minimum probability for a locus to be considered active. (default: None)

--no-alt-contigs

Ignore commonly known alternate contigs. (default: None)

--genotype-germline-sites

Call all apparent germline site even though they will ultimately be filtered. (default: None)

--genotype-pon-sites

Call sites in the PoN even though they will ultimately be filtered. (default: None)

--force-call-filtered-alleles

Force-call filtered alleles included in the resource specified by --alleles. (default: None)

--tumor-name TUMOR_NAME

Name of the sample for tumor reads. This MUST match the SM tag in the tumor BAM file. (default: None)

Option is required.

--normal-name NORMAL_NAME

Name of the sample for normal reads. If specified, this MUST match the SM tag in the normal BAM file. (default: None)

-L INTERVAL, --interval INTERVAL

Interval within which to call the variants from the BAM/CRAM file. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

Performance Options:

--mutect-low-memory

Use low memory mode in mutect caller. (default: None)

--run-partition

Turn on partition mode; divides genome into multiple partitions and runs 1 process per partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--num-htvc-threads NUM_HTVC_THREADS

Number of CPU threads to use. (default: 5)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the

PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)
--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

pacbio_germline (Beta)

Run the germline variant tool to generate BAM and variants on long read sequences using minimap2 for alignment as well as the DeepVariant variant caller.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ $ pbrun pacbio_germline \ --ref /workdir/${REFERENCE_FILE} \ --
in-fq /workdir/${INPUT_FASTQ} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-
variants /outputdir/${OUTPUT_VCF}
```

Compatible CPU-based minimap2, GATK4, and Google DeepVariant Commands

The commands below are the minimap2-v2.26, GATK4, and Google DeepVariant counterpart of the Clara Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# Run minimap2 and pipe the output to create a sorted BAM. $ minimap2 -ax map-
pbmm2 \ <INPUT_DIR>/${REFERENCE_FILE} \ <INPUT_DIR>/${INPUT_FASTQ} | \ gatk
SortSam \ --java-options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin
\ -O cpu.bam \ --SORT_ORDER coordinate # Run deepvariant BIN_VERSION="1.6.1"
sudo docker run \ -v "${PWD}":"/input" \ -v "${PWD}/output":"/output" \ -v
"${PWD}/Ref":"/reference" \ google/deepvariant:"${BIN_VERSION}" \
/opt/deepvariant/bin/run_deepvariant \ --model_type PACBIO \ --ref
/reference/${REFERENCE_FILE} \ --reads cpu.bam \ --output_vcf
/output/"${OUTPUT_VCF_FILE}" \ --num_shards $(nproc) \ --
make_examples_extra_args "ws_use_window_selector_model=true"
```

Please note that two changes must be made to the baseline minimap2 code in order to match the results exactly:

Firstly, a new preset must be made in `options.c` in the `mm_set_opt` function that tries to replicate the preset of pbmm2 by setting these parameters as a new preset named "map-pbmm2":

```
io->k = 19; io->w = 10; io->batch_size = 0x7fffffffffffffffL; // always build a uni-part
index mo->flag |= MM_I_HPC; mo->flag |= MM_F_CIGAR; mo->flag |=
MM_F_LONG_CIGAR; mo->flag |= MM_F_EQX; mo->flag |= MM_F_SOFTCLIP; mo-
>flag |= MM_F_NO_PRINT_2ND; mo->flag |= MM_F_HARD_MLEVEL; mo->mask_level
```

```
= 0; mo->e2 = 1; mo->zdrop = 400; mo->a = 2; mo->b = 5; mo->q = 5; mo->q2 = 56;
mo->e = 4; mo->zdrop_inv = 50; mo->bw = 2000;
```

Secondly, a fix must be made to the baseline KSW2 code to round the loop fission start and end points by changing them to `st` and `en` respectively. If the start point (`st0`) is a number below 16, but greater than 0, its scoring values will not be initialized correctly, but will still be used later when computing the actual alignment. This can be fixed by rounding the start and end points to multiples of 16.

To make this fix, change the following code in `ksw2_extd2_sse.c`:

```
// loop fission: set scores first if (!(flag & KSW_EZ_GENERIC_SC)) { for (t = st0; t <= en0;
t += 16) { __m128i sq, st, tmp, mask; sq = _mm_loadu_si128((__m128i*)&sf[t]); st =
_mm_loadu_si128((__m128i*)&qrr[t]); mask = _mm_or_si128(_mm_cmpeq_epi8(sq,
m1_), _mm_cmpeq_epi8(st, m1_)); tmp = _mm_cmpeq_epi8(sq, st); #ifdef __SSE4_1__
tmp = _mm_blendv_epi8(sc_mis_, sc_mch_, tmp); tmp = _mm_blendv_epi8(tmp,
sc_N_, mask); #else tmp = _mm_or_si128(_mm_andnot_si128(tmp, sc_mis_),
_mm_and_si128(tmp, sc_mch_)); tmp = _mm_or_si128(_mm_andnot_si128(mask,
tmp), _mm_and_si128(mask, sc_N_)); #endif _mm_storeu_si128((__m128i*)((int8_t*)s
+ t), tmp); } } else { for (t = st0; t <= en0; ++t) ((uint8_t*)s)[t] = mat[sf[t] * m + qrr[t]]; }
```

Fixed version that uses `lf_start` and `lf_en`:

```
// loop fission: set scores first int lf_start = st, lf_en = en; if (!(flag &
KSW_EZ_GENERIC_SC)) { for (t = lf_start; t <= lf_en; t += 16) { __m128i sq, st, tmp,
mask; sq = _mm_loadu_si128((__m128i*)&sf[t]); st =
_mm_loadu_si128((__m128i*)&qrr[t]); mask = _mm_or_si128(_mm_cmpeq_epi8(sq,
```

> m1_), _mm_cmpeq_epi8(st, m1_)); tmp = _mm_cmpeq_epi8(sq, st); *#ifdef __SSE4_1__* tmp = _mm_blendv_epi8(sc_mis_, sc_mch_, tmp); tmp = _mm_blendv_epi8(tmp, sc_N_, mask); *#else* tmp = _mm_or_si128(_mm_andnot_si128(tmp, sc_mis_), _mm_and_si128(tmp, sc_mch_)); tmp = _mm_or_si128(_mm_andnot_si128(mask, tmp), _mm_and_si128(mask, sc_N_)); *#endif* _mm_storeu_si128((__m128i*)((int8_t*)s + t), tmp); } } else { for (t = lf_start; t <= lf_en; ++t) ((uint8_t*)s)[t] = mat[sf[t] * m + qrr[t]]; }

Models for additional GPUs

See the DeepVariant Models for additional GPUs section for instructions on downloading and using model files for additional GPUs.

pacbio_germline Reference

Run the germline pipeline from FASTQ to VCF by aligning long read sequences with minimap2 and using a deep neural network analysis.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--index INDEX

Path to a minimizer index file generated by vanilla minimap2 to reduce indexing time. (default: None)

--in-fq IN_FQ

Path to a query sequence file in fastq or fastq.gz format. (default: None)

Option is required.

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--pb-model-file PB_MODEL_FILE

Path to a non-default parabricks model file for deepvariant. (default: None)

--out-recal-file OUT_RECAL_FILE

Path of the report file after Base Quality Score Recalibration. (default: None)

--out-bam OUT_BAM

Path of BAM file after Marking Duplicates. (default: None)

Option is required.

--out-variants OUT_VARIANTS

Path of the vcf/gvcf/gvcf.gz file after variant calling. (default: None)

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

Path of a duplicate metrics file after Marking Duplicates. (default: None)

--proposed-variants PROPOSED_VARIANTS

Path of the VCF file, which has proposed variants for the make examples stage. (default: None)

Tool Options:

--preset PRESET

Which preset to apply. Possible values are {map-pbmm2,map-hifi,map-ont}. (default: map-pbmm2)

--eqx

Write =/X CIGAR operators. (default: None)

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times (e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000"). (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR after generating sorted BAM. This option requires both --knownSites and --out-recal-file input parameters. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of fastq files in this run. The ID and PU tags will consist of this prefix and an identifier, that will be unique for a pair of fastq files. (default: None)

--disable-use-window-selector-model

Change the window selector model from Allele Count Linear to Variant Reads. This option will increase the accuracy and runtime. (default: None)

--gvcf

Generate variant calls in .gvcf Format. (default: None)

--norealign-reads

Do not locally realign reads before calling variants. Reads longer than 500 bp are never realigned. (default: None)

--sort-by-haplotypes

Reads are sorted by haplotypes (using HP tag). (default: None)

--keep-duplicates

Keep reads that are duplicate. (default: None)

--vsc-min-count-snps VSC_MIN_COUNT_SNPS

SNP alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-count-indels VSC_MIN_COUNT_INDELS

Indel alleles occurring at least this many times in the AlleleCount will be advanced as candidates. (default: 2)

--vsc-min-fraction-snps VSC_MIN_FRACTION_SNPS

SNP alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: 0.12)

--vsc-min-fraction-indels VSC_MIN_FRACTION_INDELS

Indel alleles occurring at least this fraction of all counts in the AlleleCount will be advanced as candidates. (default: None)

--min-mapping-quality MIN_MAPPING_QUALITY

By default, reads with any mapping quality are kept. Setting this field to a positive integer i will only keep reads that have a MAPQ >= i. Note this only applies to aligned reads. (default: 5)

--min-base-quality MIN_BASE_QUALITY

Minimum base quality. This option enforces a minimum base quality score for alternate alleles. Alternate alleles will only be considered if all bases in the allele have a quality greater than min_base_quality. (default: 10)

--alt-aligned-pileup ALT_ALIGNED_PILEUP

Value can be one of [none, diff_channels]. Include alignments of reads against each candidate alternate allele in the pileup image. (default: None)

--variant-caller VARIANT_CALLER

Value can be one of [VERY_SENSITIVE_CALLER, VCF_CANDIDATE_IMPORTER]. The caller to use to make examples. If you use VCF_CANDIDATE_IMPORTER, it implies force calling. Default is VERY_SENSITIVE_CALLER. (default: None)

--add-hp-channel

Add another channel to represent HP tags per read. (default: None)

--parse-sam-aux-fields

Auxiliary fields of the BAM/CRAM records are parsed. If either --sort-by-haplotypes or --add-hp-channel is set, then this option must also be set. (default: None)

--use-wes-model

If passed, the WES model file will be used. Only used in shortread mode. (default: None)

--include-med-dp

If True, include MED_DP in the output gVCF records. (default: None)

--normalize-reads

If True, allele counter left align INDELs for each read. (default: None)

--pileup-image-width PILEUP_IMAGE_WIDTH

Pileup image width. Only change this if you know your model supports this width. (default: 221)

--channel-insert-size

If True, add insert_size channel into pileup image. By default, this parameter is true in WGS and WES mode. (default: None)

--no-channel-insert-size

If True, don't add insert_size channel into the pileup image. (default: None)

--max-read-size-512

Allow deepvariant to run on reads of size 512bp. The default size is 320 bp. (default: None)

--prealign-helper-thread

Use an extra thread for the pre-align step. This parameter is more useful when --max-reads-size-512 is set. (default: None)

--track-ref-reads

If True, allele counter keeps track of reads supporting ref. By default, allele counter keeps a simple count of the number of reads supporting ref. (default: None)

--phase-reads

Calculate phases and add HP tag to all reads automatically. (default: None)

--dbg-min-base-quality DBG_MIN_BASE_QUALITY

Minimum base quality in a k-mer sequence to consider. (default: 15)

--ws-min-windows-distance WS_MIN_WINDOWS_DISTANCE

Minimum distance between candidate windows for local assembly (default: 80)

--channel-gc-content

If True, add gc_content channel into pileup image (default: None)

--channel-hmer-deletion-quality

If True, add hmer deletion quality channel into pileup image (default: None)

--channel-hmer-insertion-quality

If True, add hmer insertion quality channel into pileup image (default: None)

--channel-non-hmer-insertion-quality

If True, add non-hmer insertion quality channel into pileup image (default: None)

--skip-bq-channel

If True, ignore base quality channel. (default: None)

--aux-fields-to-keep AUX_FIELDS_TO_KEEP

Comma-delimited list of auxiliary BAM fields to keep. Values can be [HP, tp, t0] (default: HP)

--vsc-min-fraction-hmer-indels VSC_MIN_FRACTION_HMER_INDELS

Hmer Indel alleles occurring at least this be advanced as candidates. Use this threshold if hmer and non-hmer indels should be treated differently (Ultima reads)Default will use the same threshold for hmer and non-hmer indels, as defined in vsc_min_fraction_indels. (default: None)

--vsc-turn-on-non-hmer-ins-proxy-support

Add read-support from soft-clipped reads and other non-hmer insertion alleles,to the most frequent non-hmer insertion allele. (default: None)

--consider-strand-bias

If True, expect SB field in calls and write it to vcf (default: None)

--p-error P_ERROR

Basecalling error for reference confidence model. (default: 0.001)

--channel-ins-size

If true, add another channel to represent size of insertions. (good for flow-based sequencing) (default: None)

--max-ins-size MAX_INS_SIZE

Max insertion size for ins_size_channel, larger insertions will look like max (have max intensity) (default: 10)

--disable-group-variants

If using vcf_candidate_importer and multi-allelic sites are split across multiple lines in VCF, set to True so that variants are not grouped when transforming CallVariantsOutput to Variants. (default: None)

--filter-reads-too-long

Ignore all input bam reads with size > 512bp (default: None)

--haploid-contigs HAPLOID_CONTIGS

Optional list of non autosomal chromosomes. For all listed chromosomes HET probabilities are not considered. (default: None)

Performance Options:

--num-threads NUM_THREADS

Number of processing threads. (default: 128)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster

while option 3 provides a better compression ratio. (default=0) (default: None)
--gpusort

Use GPUs to accelerate sorting. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--low-memory

Use low memory mode (default: None)

--num-cpu-threads-per-stream NUM_CPU_THREADS_PER_STREAM

Number of CPU threads to use per stream. (default: 6)

--num-streams-per-gpu NUM_STREAMS_PER_GPU

Number of streams to use per GPU. (default: 2)

--run-partition

Divide the whole genome into multiple partitions and run multiple processes at the same time, each on one partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--max-reads-per-partition MAX_READS_PER_PARTITION

The maximum number of reads per partition that are considered before following processing such as sampling and realignment. (default: 1500)

--partition-size PARTITION_SIZE

The maximum number of basepairs allowed in a region before splitting it into multiple smaller subregions. (default: 1000)

--read-from-tmp-dir

Running variant caller reading from bin files generated by Aligner and sort. Run postsort in parallel. This option will increase device memory usage. (default: None)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

postpon

Annotate variants based on a PON file and modify the "INFO" field of the input VCF file. This is the post process of calling `--pon` in mutect.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun postpon \ --in-vcf /workdir/${INPUT_VCF} \ --in-pon-file
/workdir/${INPUT_PON_VCF} \ --out-vcf /outputdir/${OUTPUT_VCF}
```

postpon Reference

Annotate variants based on a PON file

Input/Output file options

--in-vcf IN_VCF

Path to the input vcf file. (default: None)

Option is required.

--in-pon-file IN_PON_FILE

Path to the input PON file in vcf.gz format with its tabix index. (default: None)

Option is required.

--out-vcf OUT_VCF

Output annotated VCF file. (default: None)

Option is required.

Options specific to this tool

(none)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

prepon

Generate an index for a PON file. This is a prerequisite for using the "--pon" option in mutectcaller.

`prepon` requires that the Contig field be present in the header of the input `.vcf.gz` file in order to do memory allocation at the start of execution. This field should include both the chromosome name and length:

```
##contig=<ID=chr1,length=248956422> ##contig=<ID=chr2,length=242193529>
##contig=<ID=chr3,length=198295559> ...
```

If your input `.vcf.gz` file does not include these value, run this command to update the header before running `prepon`:

```
$ bcftools reheader --fai YOUR_REFERENCE_FILE.fa.fai INPUT_PON.vcf.gz >
UPDATED_PON.vcf.gz
```

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun prepon \ --in-pon-file /workdir/${INPUT_PON_VCF}
```

prepon Reference

Build the index for a PON file; this is a prerequisite for mutect pon

Input/Output file options

--in-pon-file IN_PON_FILE

Path to the input PON file in vcf.gz format with its tabix index. (default: None)

Option is required.

Options specific to this tool

(none)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

rna_fq2bam

This tool is the equivalent of fq2bam for RNA-Seq samples, receiving inputs in FASTQ format, performing alignment with the splice-aware STAR algorithm, optionally marking of duplicate reads, and outputting an aligned BAM file ready for variant and fusion calling.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun rna_fq2bam \ --in-fq /workdir/${INPUT_FASTQ_1}
/workdir/${INPUT_FASTQ_2} \ --genome-lib-dir
/workdir/${PATH_TO_GENOME_LIBRARY}/ \ --output-dir
/outputdir/${PATH_TO_OUTPUT_DIRECTORY} \ --ref /workdir/${REFERENCE_FILE} \ --
out-bam /outputdir/${OUTPUT_BAM} \ --read-files-command zcat
```

Compatible CPU Command

The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# STAR Alignment $ ./STAR \ --genomeDir
<INPUT_DIR>/${PATH_TO_GENOME_LIBRARY} \ --readFilesIn
<INPUT_DIR>/${INPUT_FASTQ_1} <INPUT_DIR>/${INPUT_FASTQ_2} \ --
outFileNamePrefix <OUTPUT_DIR>/${PATH_TO_OUTPUT_DIRECTORY}/ \ --
outSAMtype BAM SortedByCoordinate \ --readFilesCommand zcat # Mark Duplicates
$ gatk MarkDuplicates \ --java-options -Xmx30g \ -I Aligned.sortedByCoord.out.bam
\# This filename is determined by STAR. -O
<OUTPUT_DIR>/${NAME_OF_OUTPUT_BAM_FILE} \ -M metrics.txt
```

Note

Make sure you have the same version of STAR installed that was used to build the genome index.

The Parabricks version of STAR is compatible with the 2.7.2a CPU-only version of STAR.

rna_fq2bam Reference

Run RNA-seq data through the fq2bam pipeline. It will run STAR aligner, co- ordinate sorting and mark duplicates.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.
--in-fq [IN_FQ ...]

Path to the pair-ended FASTQ files followed by optional read groups with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The files must be in fastq or fastq.gz format. All sets of inputs should have a read group; otherwise, none should have a read group, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-fq [IN_SE_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-se-fq sampleX_1.fastq.gz --in-se-fq sampleX_2.fastq.gz . Example 2: --in-se-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:unit1" --in-se-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:sample\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--genome-lib-dir GENOME_LIB_DIR

Path to a genome resource library directory. The indexing required to run STAR should be completed by the user beforehand. (default: None)

Option is required.

--output-dir OUTPUT_DIR

Path to the directory that will contain all of the generated files. (default: None)

Option is required.

--out-bam OUT_BAM

Path of the output BAM file. (default: None)

Option is required.

Tool Options:

--out-prefix OUT_PREFIX

Prefix filename for output data. (default: None)

--read-files-command READ_FILES_COMMAND

Command line to execute for each of the input files. This command should generate FASTA or FASTQ text and send it to stdout: For example, zcat to uncompress .gz files, bzcat to uncompress .bz2 files, etc. (default: None)

--read-group-sm READ_GROUP_SM

SM tag for read groups in this run. (default: None)

--read-group-lb READ_GROUP_LB

LB tag for read groups in this run. (default: None)

--read-group-pl READ_GROUP_PL

PL tag for read groups in this run. (default: None)

--read-group-id-prefix READ_GROUP_ID_PREFIX

Prefix for the ID and PU tags for read groups in this run. This prefix will be used for all pairs of FASTQ files in this run. The ID and PU tags will consist of this prefix and an identifier that will be unique for a pair of FASTQ files. (default: None)

--num-sa-bases NUM_SA_BASES

Length (bases) of the SA pre-indexing string. Longer strings will use more memory, but allow for faster searches. A value between 10 and 15 is recommended. For small genomes, the parameter must be scaled down to min(14, log2(GenomeLength)/2 - 1). (default: 14)

--max-intron-size MAX_INTRON_SIZE

Maximum align intron size. If this value is 0, the maximum size will be determined by (2^winBinNbits)*winAnchorDistNbins. (default: 0)

--min-intron-size MIN_INTRON_SIZE

Minimum align intron size. Genomic gap is considered intron if its length is greater than or equal to this value, otherwise it is considered Deletion. (default: 21)

--min-match-filter MIN_MATCH_FILTER

Minimum number of matched bases required for alignment output. (default: 0)

--min-match-filter-normalized MIN_MATCH_FILTER_NORMALIZED

Same as --min-match-filter, but normalized to the read length (sum of the mate lengths for paired-end reads). (default: 0.66)

--out-filter-intron-motifs OUT_FILTER_INTRON_MOTIFS

Type of filter alignment using its motifs. This string can be "None" for no filtering, "RemoveNoncanonical" for filtering out alignments that contain non-canonical junctions, or "RemoveNoncanonicalUnannotated" for filtering out alignments that contain non-canonical unannotated junctions when using the annotated splice junctions database. The annotated non-canonical junctions will be kept. (default: None)

--max-out-filter-mismatch MAX_OUT_FILTER_MISMATCH

Maximum number of mismatches allowed for an alignment to be output. (default: 10)

--max-out-filter-mismatch-ratio MAX_OUT_FILTER_MISMATCH_RATIO

Maximum ratio of mismatches to mapped length allowed for an alignment to be output. (default: 0.3)

--max-out-filter-multimap MAX_OUT_FILTER_MULTIMAP

Maximum number of loci the read is allowed to map to for all alignments to be output. Otherwise, no alignments will be output and the read will be counted as "mapped to too many loci" in the Log.final.out. (default: 10)

--out-reads-unmapped OUT_READS_UNMAPPED

Type of output of unmapped and partially mapped (i.e. mapped only one mate of a paired-end read) reads in separate file(s). This string can be "None" for no output or "Fastx" for output in separate FASTA/FASTQ files, Unmapped.out.mate1/2. (default: None)

--out-sam-unmapped OUT_SAM_UNMAPPED

Type of output of unmapped reads in SAM format. The string can be "None" to produce no output, "Within" to output unmapped reads within the main SAM file, "KeepPairs" to produce no output (with unmapped mates will be recorded for each alignment), or "Within_KeepPairs" to output unmapped reads within the main SAM file (with unmapped mates recorded for each alignment). (default: None)

--out-sam-attributes OUT_SAM_ATTRIBUTES [OUT_SAM_ATTRIBUTES ...]

A string of SAM attributes in the order desired for the output SAM. The string can contain any combination of the following attributes: {NH, HI, AS, nM, NM, MD, jM, jI, XS, MC, ch}. Alternatively, the string can be "None" for no attributes, "Standard" for the attributes {NH, HI, AS, nM}, or "All" for the attributes {NH, HI, AS, nM, NM, MD, jM, jI, MC, ch} (e.g. "--outSAMattributes NH nM jI XS ch"). (default: Standard)

--out-sam-strand-field OUT_SAM_STRAND_FIELD

Cufflinks-like strand field flag. The string can be "None" for no flag or "intronMotif" for the strand derived from the intron motif. Reads with inconsistent and/or non-canonical introns will be filtered out. (default: None)

--out-sam-mode OUT_SAM_MODE

SAM output mode. The string can be "None" for no SAM output, "Full" for full SAM output, or "NoQS" for full SAM output without quality scores. (default: Full)

--out-sam-mapq-unique OUT_SAM_MAPQ_UNIQUE

The MAPQ value for unique mappers. Must be in the range [0, 255]. (default: 255)

--min-score-filter MIN_SCORE_FILTER

Minimum score required for alignment output, normalized to the read length (i.e. the sum of mate lengths for paired-end reads). (default: 0.66)

--min-spliced-mate-length MIN_SPLICED_MATE_LENGTH

Minimum mapped length for a read mate that is spliced and normalized to the mate length. Must be greater than 0. (default: 0.66)

--max-junction-mismatches MAX_JUNCTION_MISMATCHES MAX_JUNCTION_MISMATCHES MAX_JUNCTION_MISMATCHES MAX_JUNCTION_MISMATCHES

Maximum number of mismatches for stitching of the splice junctions. A limit must be specified for each of the following: (1) non-canonical motifs, (2) GT/AG and CT/AC motif, (3) GC/AG and CT/GC motif, (4) AT/AC and GT/AT motif. To indicate no limit for any of the four options, use -1. (default: [0, -1, 0, 0])

--max-out-read-size MAX_OUT_READ_SIZE

Maximum size of the SAM record (bytes) for one read. Recommended value: >(2*(LengthMate1+LengthMate2+100)*o utFilterMultimapNmax. Must be greater than 0. (default: 100000)

--max-alignments-per-read MAX_ALIGNMENTS_PER_READ

Maximum number of different alignments per read to consider. Must be greater than 0. (default: 10000)

--score-gap SCORE_GAP

Splice junction penalty (independent of intron motif). (default: 0)

--seed-search-start SEED_SEARCH_START

Defines the search start point through the read. The read split pieces will not be longer than this value. Must be greater than 0. (default: 50)

--max-bam-sort-memory MAX_BAM_SORT_MEMORY

Maximum available RAM (bytes) for sorting BAM. If this value is 0, it will be set to the genome index size. Must be greater than or equal to 0. (default: 0)

--align-ends-type ALIGN_ENDS_TYPE

Type of read ends alignment. Can be one of two options: "Local" will perform a standard local alignment with soft-clipping allowed; "EndToEnd" will force an end-to-end read

alignment with no soft-clipping. (default: Local)
--align-insertion-flush ALIGN_INSERTION_FLUSH

Flush ambiguous insertion positions. The string can be "None" to not flush insertions or "Right" to flush insertions to the right. (default: None)

--max-align-mates-gap MAX_ALIGN_MATES_GAP

Maximum gap between two mates. If 0, the max intron gap will be determined by (2^winBinNbits)*winAnchorDistNbins. (default: 0)

--min-align-spliced-mate-map MIN_ALIGN_SPLICED_MATE_MAP

Minimum mapped length for a read mate that is spliced. Must be greater than or equal to 0. (default: 0)

--max-collapsed-junctions MAX_COLLAPSED_JUNCTIONS

Maximum number of collapsed junctions. Must be greater than 0. (default: 1000000)

--min-align-sj-overhang MIN_ALIGN_SJ_OVERHANG

Minimum overhang (i.e. block size) for spliced alignments. Must be greater than 0. (default: 5)

--min-align-sjdb-overhang MIN_ALIGN_SJDB_OVERHANG

Minimum overhang (i.e. block size) for annotated (sjdb) spliced alignments. Must be greater than 0. (default: 3)

--sjdb-overhang SJDB_OVERHANG

Length of the donor/acceptor sequence on each side of the junctions. Ideally, this value should be equal to mate_length - 1. Must be greater than 0. (default: 100)

--min-chim-overhang MIN_CHIM_OVERHANG

Minimum overhang for the Chimeric.out.junction file. Must be greater than or equal to 0. (default: 20)

--min-chim-segment MIN_CHIM_SEGMENT

Minimum chimeric segment length. If it is set to 0, there will be no chimeric output. Must be greater than or equal to 0. (default: 0)

--max-chim-multimap MAX_CHIM_MULTIMAP

Maximum number of chimeric multi-alignments. If it is set to 0, the old scheme for chimeric detection, which only considered unique alignments, will be used. Must be greater than or equal to 0. (default: 0)

--chim-multimap-score-range CHIM_MULTIMAP_SCORE_RANGE

The score range for multi-mapping chimeras below the best chimeric score. This option only works with --max-chim-multimap > 1. Must be greater than or equal to 0. (default: 1)

--chim-score-non-gtag CHIM_SCORE_NON_GTAG

The penalty for a non-GT/AG chimeric junction. (default: -1)

--min-non-chim-score-drop MIN_NON_CHIM_SCORE_DROP

To trigger chimeric detection, the drop in the best non-chimeric alignment score with respect to the read length has to be smaller than this value. Must be greater than or equal to 0. (default: 20)

--out-chim-format OUT_CHIM_FORMAT

Formatting type for the Chimeric.out.junction file. Possible types are {0, 1}. If type 0, there will be no comment lines/headers. If type 1, there will be comment lines at the end of the file: command line and Nreads: total, unique, multi. (default: 0)

--two-pass-mode TWO_PASS_MODE

Two-pass mapping mode. The string can be "None" for one-pass mapping or "Basic" for basic two-pass mapping, with all first pass junctions inserted into the genome indices on the fly. (default: None)

--out-chim-type OUT_CHIM_TYPE

Type of chimeric output. This string can be "Junctions" for Chimeric.out.junction, "WithinBAM" for main aligned BAM files (Aligned.*.bam), "WithinBAM_HardClip" for hard-clipping in the CIGAR for supplemental chimeric alignments, or "WithinBAM_SoftClip" for soft-clipping in the CIGAR for supplemental chimeric alignments. (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--read-name-separator READ_NAME_SEPARATOR [READ_NAME_SEPARATOR ...]

Character(s) separating the part of the read names that will be trimmed in output (read name after space is always trimmed). (default: /)

Performance Options:

--num-threads NUM_THREADS

Number of running worker threads per GPU. (default: 4)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

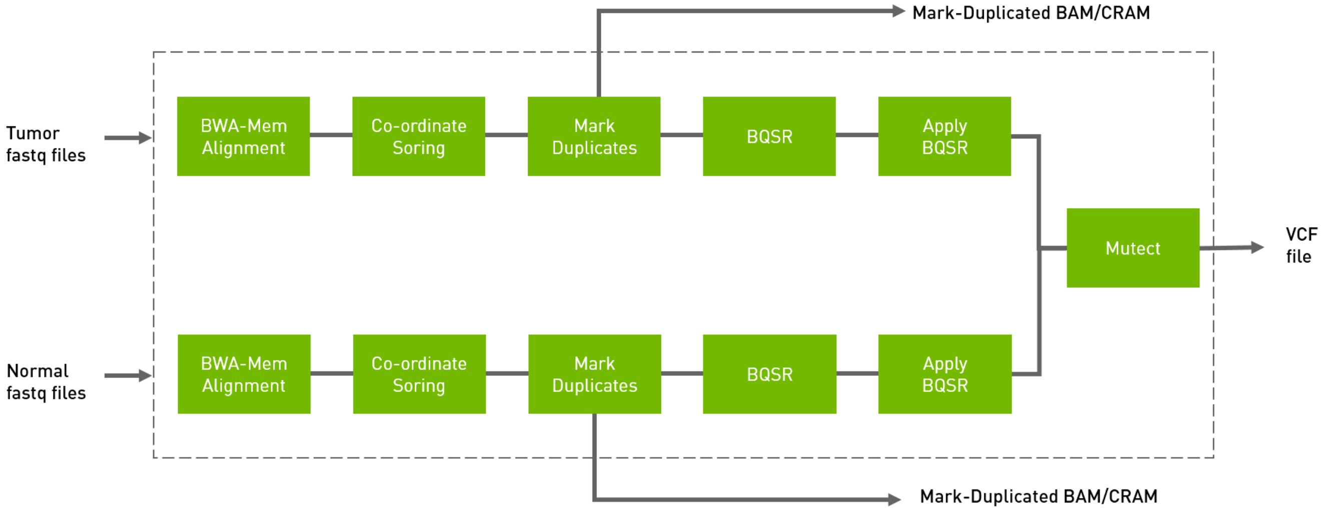
> **Note**
>
> The *--in-fq* option takes the names of two FASTQ files, optionally followed by a quoted read group. The FASTQ filenames must not start with a hyphen.

somatic (Somatic Variant Caller)

Run a somatic variant workflow.

The somatic tool processes the tumor FASTQ files, and optionally normal FASTQ files and knownSites files, and generates tumor or tumor/normal analysis. The output is in VCF format.

Internally the somatic tool runs several other Parabricks tools, thereby simplifying your work flow.



Quick Start

```
# The command line below will run tumor-only analysis. # 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 \ --workdir
/workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun somatic \ --ref
/workdir/${REFERENCE_FILE} \ --in-tumor-fq /workdir/${INPUT_FASTQ_1}
/workdir/${INPUT_FASTQ_2} \ --bwa-options="-Y" \ --out-vcf
/outputdir/${OUTPUT_VCF} \ --out-tumor-bam /outputdir/${OUTPUT_BAM} # The
command line below will run tumor-normal analysis. # This command assumes all the
inputs are in INPUT_DIR and all the outputs go to OUTPUT_DIR. docker run --rm --gpus
all --volume INPUT_DIR:/workdir --volume OUTPUT_DIR:/outputdir \ --workdir
/workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun somatic \ --ref
/workdir/${REFERENCE_FILE} \ --knownSites /workdir/${KNOWN_SITES_FILE} \ --in-
tumor-fq /workdir/${INPUT_TUMOR_FASTQ_1} /workdir/${INPUT_TUMOR_FASTQ_2}
"@RG\tID:sm_tumor_rg1\tLB:lib1\tPL:bar\tSM:sm_tumor\tPU:sm_tumor_rg1" \ --
bwa-options="-Y" \ --out-vcf /outputdir/${OUTPUT_VCF} \ --out-tumor-bam
/outputdir/${OUTPUT_TUMOR_BAM} \ --out-tumor-recal-file
/outputdir/${OUTPUT_RECAL_FILE} \ --in-normal-fq
/workdir/${INPUT_NORMAL_FASTQ_1} /workdir/${INPUT_NORMAL_FASTQ_2}
"@RG\tID:sm_normal_rg1\tLB:lib1\tPL:bar\tSM:sm_normal\tPU:sm_normal_rg1" \ --
out-normal-bam /outputdir/${OUTPUT_NORMAL_BAM}
```

Compatible CPU Command

```
# The commands below will run tumor-normal analysis. # # Run bwa mem on the tumor
FASTQ files then sort the BAM by coordinates. $ bwa mem \ -t 32 \ -K 10000000 \ -Y \ -R
'@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \
${REFERENCE_FILE} ${TUMOR_FASTQ_1} ${TUMOR_FASTQ_2} | \ gatk SortSam \ --
java-options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O
tumor_cpu.bam \ --SORT_ORDER coordinate # Mark duplicates. $ gatk
MarkDuplicates \ --java-options -Xmx30g \ -I tumor_cpu.bam \ -O
tumor_mark_dups_cpu.bam \ -M tumor_metrics.txt # Generate a BQSR report. $ gatk
BaseRecalibrator \ --java-options -Xmx30g \ --input tumor_mark_dups_cpu.bam \ --
output ${OUTPUT_TUMOR_RECAL_FILE} \ --known-sites ${KNOWN_SITES_FILE} \ --
reference ${REFERENCE_FILE} # Apply the BQSR report. $ gatk ApplyBQSR \ --java-
options -Xmx30g \ -R ${REFERENCE_FILE} \ -I tumor_cpu.bam \ --bqsr-recal-file
${TUMOR_OUTPUT_RECAL_FILE} \ -O ${OUTPUT_TUMOR_BAM} # Now repeat all the
above steps, only with the normal FASTQ data. $ bwa mem \ -t 32 \ -K 10000000 \ -Y \ -
R '@RG\tID:sample_rg1\tLB:lib1\tPL:bar\tSM:sample\tPU:sample_rg1' \
${REFERENCE_FILE} ${NORMAL_FASTQ_1} ${NORMAL_FASTQ_2} | \ gatk SortSam \ --
java-options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O
normal_cpu.bam \ --SORT_ORDER coordinate # Mark duplicates. $ gatk
MarkDuplicates \ --java-options -Xmx30g \ -I normal_cpu.bam \ -O
normal_mark_dups_cpu.bam \ -M normal_metrics.txt # Generate a BQSR report. $
gatk BaseRecalibrator \ --java-options -Xmx30g \ --input
normal_mark_dups_cpu.bam \ --output ${OUTPUT_NORMAL_RECAL_FILE} \ --known-
sites ${KNOWN_SITES_FILE} \ --reference ${REFERENCE_FILE} # Apply the BQSR
report. $ gatk ApplyBQSR \ --java-options -Xmx30g \ -R ${REFERENCE_FILE} \ -I
normal_cpu.bam \ --bqsr-recal-file ${OUTPUT_NORMAL_RECAL_FILE} \ -O
${OUTPUT_NORMAL_BAM} # Finally, run Mutect2 on the normal and tumor data. $
gatk Mutect2 \ -R ${REFERENCE_FILE} \ --input ${OUTPUT_TUMOR_BAM} \ --tumor-
sample tumor \ --input ${OUTPUT_NORMAL_BAM} \ --normal-sample normal \ --
output ${OUTPUT_VCF}
```

somatic Reference

Run the tumor normal somatic pipeline from FASTQ to VCF.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--in-tumor-fq [IN_TUMOR_FQ ...]

Path to the pair-ended FASTQ files followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:20"). The files can be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one, and it will be automatically added by the pipeline. This option can be repeated multiple times. Example 1: --in-tumor-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-tumor-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz. Example 2: --in-tumor-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG ID:foo\tLB:lib1\tPL:bar\tSM:sm_tumor\tPU:unit1" --in-tumor-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG ID:foo2\tLB:lib1\tPL:bar\tSM:sm_tumor\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-se-tumor-fq [IN_SE_TUMOR_FQ ...]

Path to the single-ended FASTQ file followed by an optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one; if no read group is provided, one will be added automatically by the pipeline. This option can be repeated multiple times. Example 1: --in-se-tumor-fq sampleX_1.fastq.gz --in-se-tumor-fq sampleX_2.fastq.gz . Example 2: --in-se-tumor-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:tumor\tPU:unit1" --in-se-tumor-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:tumor\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--in-normal-fq [IN_NORMAL_FQ ...]

Path to the pair-ended FASTQ files followed by an optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:20"). The files must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one; if no read group is provided, one will be automatically added by the pipeline. This option can be

repeated multiple times. Example 1: --in-normal-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz --in-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz . Example 2: --in-normal-fq sampleX_1_1.fastq.gz sampleX_1_2.fastq.gz "@RG ID:foo\tLB:lib1\tPL:bar\tSM:sm_normal\tPU:unit1" --in-normal-fq sampleX_2_1.fastq.gz sampleX_2_2.fastq.gz "@RG ID:foo2\tLB:lib1\tPL:bar\tSM:sm_normal\tPU:unit2". For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)
--in-se-normal-fq [IN_SE_NORMAL_FQ ...]

Path to the single-ended FASTQ file followed by optional read group with quotes (Example: "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:sample\tPU:foo"). The file must be in fastq or fastq.gz format. Either all sets of inputs have a read group, or none should have one; if no read group is provided, one will be added automatically by the pipeline. This option can be repeated multiple times. Example 1: --in-se-normal-fq sampleX_1.fastq.gz --in-se-normal-fq sampleX_2.fastq.gz . Example 2: --in-se-normal-fq sampleX_1.fastq.gz "@RG\tID:foo\tLB:lib1\tPL:bar\tSM:normal\tPU:unit1" --in-se-normal-fq sampleX_2.fastq.gz "@RG\tID:foo2\tLB:lib1\tPL:bar\tSM:normal\tPU:unit2" . For the same sample, Read Groups should have the same sample name (SM) and a different ID and PU. (default: None)

--knownSites KNOWNSITES

Path to a known indels file. The file must be in vcf.gz format. This option can be used multiple times. (default: None)

--interval-file INTERVAL_FILE

Path to an interval file in one of these formats: Picard-style (.interval_list or .picard), GATK-style (.list or .intervals), or BED file (.bed). This option can be used multiple times. (default: None)

--out-vcf OUT_VCF

Path of the VCF file after Variant Calling. (default: None)

Option is required.

--out-tumor-bam OUT_TUMOR_BAM

Path of the BAM file for tumor reads. (default: None)

Option is required.

--out-normal-bam OUT_NORMAL_BAM

Path of the BAM file for normal reads. (default: None)

--mutect-bam-output MUTECT_BAM_OUTPUT

File to which assembled haplotypes should be written in Mutect. (default: None)

--out-tumor-recal-file OUT_TUMOR_RECAL_FILE

Path of the report file after Base Quality Score Recalibration for tumor sample. (default: None)

--out-normal-recal-file OUT_NORMAL_RECAL_FILE

Path of the report file after Base Quality Score Recalibration for normal sample. (default: None)

--mutect-germline-resource MUTECT_GERMLINE_RESOURCE

Path of the vcf.gz germline resource file. Population vcf of germline sequencing containing allele fractions. (default: None)

--mutect-alleles MUTECT_ALLELES

Path of the vcf.gz force-call file. The set of alleles to force-call regardless of evidence. (default: None)

Tool Options:

-L INTERVAL, --interval INTERVAL

Interval within which to call bqsr from the input reads. All intervals will have a padding of 100 to get read records, and overlapping intervals will be combined. Interval files should be passed using the --interval-file option. This option can be used multiple times e.g. "-L chr1 -L chr2:10000 -L chr3:20000+ -L chr4:10000-20000". (default: None)

--bwa-options BWA_OPTIONS

Pass supported bwa mem options as one string. The current original bwa mem supported options are -M, -Y and -T e.g. --bwa-options="-M -Y" (default: None)

--no-warnings

Suppress warning messages about system thread and memory usage. (default: None)

--filter-flag FILTER_FLAG

Don't generate SAM entries in the output if the entry's flag's meet this criteria. Criteria: (flag & filter != 0) (default: 0)

--skip-multiple-hits

Filter SAM entries whose length of SA is not 0. (default: None)

--min-read-length MIN_READ_LENGTH

Skip reads below minimum read length. They will not be part of the output. (default: None)

--align-only

Generate output BAM after bwa-mem. The output will not be co-ordinate sorted or duplicates will not be marked. (default: None)

--no-markdups

Do not perform the Mark Duplicates step. Return BAM after sorting. (default: None)

--fix-mate

Add mate cigar (MC) and mate quality (MQ) tags to the output file. (default: None)

--markdups-assume-sortorder-queryname

Assume the reads are sorted by queryname for Marking Duplicates. This will mark secondary, supplementary, and unmapped reads as duplicates as well. This flag will not impact variant calling while increasing processing times. (default: None)

--markdups-picard-version-2182

Assume marking duplicates to be similar to Picard version 2.18.2. (default: None)

--monitor-usage

Monitor approximate CPU utilization and host memory usage during execution. (default: None)

--optical-duplicate-pixel-distance OPTICAL_DUPLICATE_PIXEL_DISTANCE

The maximum offset between two duplicate clusters in order to consider them optical duplicates. Ignored if --out-duplicate-metrics is not passed. (default: None)

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

Amount of padding (in base pairs) to add to each interval you are including. (default: None)

--standalone-bqsr

Run standalone BQSR. (default: None)

--max-read-length-fq2bamfast MAX_READ_LENGTH_FQ2BAMFAST

Maximum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (Argument only applies to --fq2bamfast) (default: 480)

--min-read-length-fq2bamfast MIN_READ_LENGTH_FQ2BAMFAST

Minimum read length/size (i.e., sequence length) used for bwa and filtering FASTQ input (Argument only applies to --fq2bamfast) (default: 10)

--max-mnp-distance MAX_MNP_DISTANCE

Two or more phased substitutions separated by this distance or less are merged into MNPs. (default: 1)

--mutectcaller-options MUTECTCALLER_OPTIONS

Pass supported mutectcaller options as one string. The following are currently supported original mutectcaller options: -pcr-indel-model <NONE, HOSTILE, AGGRESSIVE, CONSERVATIVE>, -max-reads-per-alignment-start <int>, (e.g. --mutectcaller-options="-pcr-indel-model HOSTILE -max-reads-per-alignment-start 30"). (default: None)

--initial-tumor-lod INITIAL_TUMOR_LOD

Log 10 odds threshold to consider pileup active. (default: None)

--tumor-lod-to-emit TUMOR_LOD_TO_EMIT

Log 10 odds threshold to emit variant to VCF. (default: None)

--pruning-lod-threshold PRUNING_LOD_THRESHOLD

Ln likelihood ratio threshold for adaptive pruning algorithm. (default: None)

--active-probability-threshold ACTIVE_PROBABILITY_THRESHOLD

Minimum probability for a locus to be considered active. (default: None)

--no-alt-contigs

Ignore commonly known alternate contigs. (default: None)

--genotype-germline-sites

Call all apparent germline site even though they will ultimately be filtered. (default: None)

--genotype-pon-sites

Call sites in the PoN even though they will ultimately be filtered. (default: None)

--force-call-filtered-alleles

Force-call filtered alleles included in the resource specified by --alleles. (default: None)

--tumor-read-group-sm TUMOR_READ_GROUP_SM

SM tag for read groups for tumor sample. (default: None)

--tumor-read-group-lb TUMOR_READ_GROUP_LB

LB tag for read groups for tumor sample. (default: None)

--tumor-read-group-pl TUMOR_READ_GROUP_PL

PL tag for read groups for tumor sample. (default: None)

--tumor-read-group-id-prefix TUMOR_READ_GROUP_ID_PREFIX

Prefix for ID and PU tag for read groups for tumor sample. This prefix will be used for all pair of tumor FASTQ files in this run. The ID and PU tag will consist of this prefix and an identifier which will be unique for a pair of FASTQ files. (default: None)

--normal-read-group-sm NORMAL_READ_GROUP_SM

SM tag for read groups for normal sample. (default: None)

--normal-read-group-lb NORMAL_READ_GROUP_LB

LB tag for read groups for normal sample. (default: None)

--normal-read-group-pl NORMAL_READ_GROUP_PL

PL tag for read groups for normal sample. (default: None)

--normal-read-group-id-prefix NORMAL_READ_GROUP_ID_PREFIX

Prefix for ID and PU tags for read groups of a normal sample. This prefix will be used for all pairs of normal FASTQ files in this run. The ID and PU tags will consist of this prefix and an identifier that will be unique for a pair of FASTQ files. (default: None)

Performance Options:

--fq2bamfast

Use fq2bamfast as the alignment tool instead of fq2bam (default: None)

--gpuwrite

Use one GPU to accelerate writing final BAM. (default: None)

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

Choose the nvCOMP DEFLATE algorithm to use with --gpuwrite. Note these options do not correspond to CPU DEFLATE options. Valid options are 0 and 3. Option 0 is faster while option 3 provides a better compression ratio. (default=0) (default: None)

--gpusort

Use GPUs to accelerate sorting and marking. (default: None)

--use-gds

Use GPUDirect Storage (GDS) to enable a direct data path for direct memory access (DMA) transfers between GPU memory and storage. Must be used concurrently with --*gpuwrite*. Please refer to Parabricks Documentation > Best Performance for information on how to set up and use GPUDirect Storage. (default: None)

--memory-limit MEMORY_LIMIT

System memory limit in GBs during sorting and postsorting. By default, the limit is half of the total system memory. (default: 62)

--low-memory

Use low memory mode (default: None)

--num-cpu-threads-per-stage NUM_CPU_THREADS_PER_STAGE

Number of CPU threads to use per stage. (default: 8)

--bwa-nstreams BWA_NSTREAMS

Number of streams per GPU to use; note: more streams increases device memory usage (Argument only applies to --fq2bamfast) (default: 4)

--bwa-cpu-thread-pool BWA_CPU_THREAD_POOL

Number of threads to devote to CPU thread pool *per GPU* (Argument only applies to --fq2bamfast) (default: 16)

--mutect-low-memory

Use low memory mode in mutect caller. (default: None)

--run-partition

Turn on partition mode; divides genome into multiple partitions and runs 1 process per partition. (default: None)

--gpu-num-per-partition GPU_NUM_PER_PARTITION

Number of GPUs to use per partition. (default: None)

--num-htvc-threads NUM_HTVC_THREADS

Number of CPU threads to use. (default: 5)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

Number of GPUs to use for a run. GPUs 0..(NUM_GPUS-1) will be used.

starfusion

Identifies candidate fusion transcripts.

This tool performs fusion calling for RNA-Seq samples, utilizing the STAR-Fusion algorithm. This requires input of a genome resource library, in accordance with the original STAR-Fusion tool, and outputs candidate fusion transcripts.

Quick Start

```
# 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 \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-
parabricks:4.3.1-1 \ pbrun starfusion \ --chimeric-junction
/workdir/${CHIMERIC_JUNCTION_INPUT} \ --genome-lib-dir
/workdir/${PATH_TO_GENOME_LIBRARY}/ \ --output-dir
/outputdir/${PATH_TO_OUTPUT_DIRECTORY}/
```

Compatible CPU Command

The command below is the CPU counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command.

```
$ ./STAR-Fusion \ --chimeric_junction <INPUT_DIR>/${CHIMERIC_JUNCTION_INPUT} \
--genome_lib_dir <INPUT_DIR>/${PATH_TO_GENOME_LIBRARY} \ --output_dir
<OUTPUT_DIR>/${PATH_TO_OUTPUT_DIRECTORY}
```

starfusion Reference

Identify candidate fusion transcripts supported by Illumina reads.

Input/Output file options

--chimeric-junction CHIMERIC_JUNCTION

Path to the Chimeric.out.junction file produced by STAR. (default: None)

Option is required.

--genome-lib-dir GENOME_LIB_DIR

Path to a genome resource library directory. For more information, visit https://github.com/STAR-Fusion/STAR-Fusion/wiki/installing-star-fusion#data-resources-required. (default: None)

Option is required.

--output-dir OUTPUT_DIR

Path to the directory that will contain all of the generated files. (default: None)

Option is required.

Tool Options:

--out-prefix OUT_PREFIX

Prefix filename for output data. (default: None)

Performance Options:

--num-threads NUM_THREADS

Number of threads for worker. (default: 4)

Common options:

--logfile LOGFILE

Path to the log file. If not specified, messages will only be written to the standard error output. (default: None)

--tmp-dir TMP_DIR

Full path to the directory where temporary files will be stored.

--with-petagene-dir WITH_PETAGENE_DIR

Full path to the PetaGene installation directory. By default, this should have been installed at /opt/petagene. Use of this option also requires that the PetaLink library has been preloaded by setting the LD_PRELOAD environment variable. Optionally set the PETASUITE_REFPATH and PGCLOUD_CREDPATH environment variables that are used for data and credentials (default: None)

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

Do not override seccomp options for docker (default: None).

--version

View compatible software versions.

Grace Hopper Superchip

Parabricks is proud to announce full optimization and support for the groundbreaking Nvidia Grace Hopper superchip. The Nvidia GH200 Grace Hopper Superchip combines the Nvidia Grace and Hopper architectures using Nvidia NVLink-C2C to deliver a CPU+GPU coherent memory model for accelerated Artificial Intelligence (AI) and High Performance Computing (HPC) applications. This integration represents a significant leap forward in genomic data analysis, allowing researchers to tackle complex analyses with unprecedented speed and efficiency. The Nvidia Grace Hopper Superchip is the first true heterogeneous accelerated platform for HPC and AI workloads. It accelerates applications with the strengths of both GPUs and CPUs while providing the simplest and most productive programming model for performance, portability, and productivity.

Key Features of the Nvidia Grace Hopper Superchip GH200

Feature	Description
Grace CPU cores (number)	Up to 72 cores
CPU LPDDR5X bandwidth (GB/s)	Up to 500GB/s
GPU HBM bandwidth (GB/s)	4TB/s HBM3, 4.9TB/s HBM3e
NVLink-C2C bandwidth (GB/s)	900GB/s total, 450GB/s per direction
CPU LPDDR5X capacity (GB)	Up to 480GB
GPU HBM capacity (GB)	96GB HBM3, 144GB HBM3e
PCIe Gen 5 Lanes	64x

By harnessing the immense computational capabilities of the Nvidia Grace Hopper Superchip, users can experience even greater acceleration and throughput for their genomic pipelines.

Documentation

All tools and pipelines from Parabricks 4.3.1-1 are now optimized and supported on the Nvidia Grace Hopper Superchip, therefore we refer the users and developers to the Tool Reference.

Performance tuning

To achieve optimal performance for all Parabricks tools on the Nvidia Grace CPU we refer the users and developers to the Grace CPU benchmarking guide. This guide will illustrate recommendations and best practices directly related to the Nvidia Grace CPU and help you realize the best possible performance for your particular system.

Get Started

- To begin using Parabricks and unleash the power of the Nvidia Grace Hopper Superchip for your genomic analyses, you can obtain the docker image running the following command:

  ```
  $ docker pull nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1.grace
  ```

  and follow the Tutorials.

- For any questions or support, please visit the Nvidia Parabricks Community. Join a vibrant community of researchers and experts to exchange ideas, seek assistance, and stay updated on the latest developments in genomic data analysis.

- To learn more about the Nvidia Grace Hopper Superchip please visit here.

Help

Frequently asked questions

1. What is NVIDIA Parabricks?

   Parabricks is a software tool for speeding up DNA and RNA analysis.

2. Where does the name come from?

   We started building small components that could be used to speed up computation much like you'd use bricks to build a house. Those "bricks" developed into larger components, which eventually became Parabricks.

3. What hardware and software do I need?

   For hardware, at least one supported GPU. How much memory you need depends on how many CPU cores you have. See Hardware Requirements for more details.

   For software, Parabricks comes as a Docker container so you'll need nvidia-docker2 and an OS that supports it. You'll also need an NVIDIA driver, version 525.60.13 or later. See Software Requirements for more details.

   That's all - because Parabricks comes as a Docker container no additional software installation is required.

4. How do I ensure that my Parabricks analysis won't terminate if I lose my SSH connection?

   This is only a problem if you're running on a remote computer - an AWS instance, for example. It's not specific to Parabricks; it can happen with any program being run on the remote computer.

   A simple option is to preface your command with nohup:

   ```
   $ nohup docker run ...options... pbrun <Your Command> &
   ```

   This will run your command in the background in a way that ignores all "hangup" signals (i.e. loss of SSH connection) and saves all output to a file called `nohup.out`.

   Another option is to use a program that supports persistent sessions such as screen or tmux.

5. Parabricks does not run when a system is initialized using Singularity containerization; Singularity does not load all `nvidia` modules.

   Do this if you see an initialization error:

   ```
   $ nvidia-modprobe -u -c=0
   ```

   This is only a concern with versions of Parabricks prior to v4.0.

6. Can I use Parabricks on my video card?

   Probably not. Parabricks requires at least 16 GB of GPU memory (24 GB for fq2bam, unless you use the `--low-memory` option), and runs on 'data center' GPUs. See the Hardware Requirements for more details.

User Forum

- We're here to help! Please reach out to us on our Developer Forums if you are having trouble using the software.

Note

- User guides and Reference manuals can be found on the NVIDIA Parabricks documentation page.

- Answers to many other FAQs can be found on the developer forum.

References

- Publications
- Third Party Software Notices
- Comparison with Baseline Tools
- End User License Agreements

Publications

How to Cite Us

If you are using Clara Parabricks in your research, please use the following to cite us:

Developer: NVIDIA
Year: 2019
Software: Clara Parabricks
URL: https://www.nvidia.com/en-us/clara/genomics/

For example:

NVIDIA (2019). Clara Parabricks ([version]). [download date]
<https://www.nvidia.com/en-us/clara/genomics/>

Academic References to NVIDIA Parabricks

- Juana G. Manuel, Hillary B. Heins, Sandra Crocker, Julie A. Neidich, Lisa Sadzewicz, Luke Tallon, Tychele N. Turner
  "High Coverage Highly Accurate Long-Read Sequencing of a Mouse Neuronal Cell Line Using the PacBio Revio Sequencer"

bioRxiv

- Sneha D. Goenka, John E. Gorzynski, Kishwar Shafin, Dianna G. Fisk, Trevor Pesout, Tanner D. Jensen, Jean Monlong, Pi-Chuan Chang, Gunjan Baid, Jonathan A. Bernstein, Jeffrey W. Christle, Karen P. Dalton, Daniel R. Garalde, Megan E. Grove, Joseph Guillory, Alexey Kolesnikov, Maria Nattestad, Maura R. Z. Ruzhnikov, Mehrzad Samadi, Ankit Sethia, Elizabeth Spiteri, Christopher J. Wright, Katherine Xiong, Tong Zhu, Miten Jain, Fritz J. Sedlazeck, Andrew Carroll, Benedict Paten, Euan A. Ashley
"Accelerated identification of disease-causing variants with ultra-rapid nanopore genome sequencing"
Nature Biotechnology 40, 1035-1041 (2022)

- Giovanna Carpi, Lev Gorenstein, Timothy T Harkins, Mehrzad Samadi, Pankaj Vats
"A GPU-accelerated compute framework for pathogen genomic variant identification to aid genomic epidemiology of infectious disease: a malaria case study"
Briefings in Bioinformatics

- Kyle A O'Connell, Zelaikha B Yosufzai, Ross A Campbell, Collin J Lobb, Haley T Engelken, Laura M Gorrell, Thad B Carlson, Josh J Catana, Dina Mikdadi, Vivien R Bonazzi, Juergen A Klenk
"Accelerating genomic workflows using NVIDIA Parabricks"
PubMed BMC Bionformatics 31 May 2023

- Peng-Chan Lin, Yi-Shan Tsai, Yu-Min Yeh, Meng-Ru Shen
"Cutting-Edge AI Technologies Meet Precision Medicine to Improve Cancer Care"
MDPI biomolecules

For more publications using Parabricks, please see Google Scholar.

Third Party Software Notices

- ABSEIL
- BCFTOOLS
- BOOST
- BWA-METH
- BWA

- BZIP2
- DEEPVARIANT
- GATK
- HTSLIB
- ISAL
- JEMALLOC
- LZ4
- MATPLOTLIB
- MINIMAP2
- NUCLEUS
- SAMTOOLS
- STAR-FUSION
- STAR
- TCMALLOC
- TENSORFLOW
- TOOLSHED
- ZLIB

Comparison with Baseline Tools

Many users want to compare output generated by NVIDIA Parabricks software with other standard tools. We recommend the following way to compare output generated by Parabricks software and the counterpart non-accelerated software.

Comparing BAM Files

GATK4 sorts the BAM files based on QNAME, FLAG, RNAME, POS, MAPQ, MRNM/RNEXT, MPOS/PNEXT, and ISIZE. If all these fields are the same for two different BAMs, they are considered equal for sorting purposes. Therefore, to compare two sorted BAMs, use the BamUtil diff tool to compare these fields:.

```
$ bam diff --in1 mark_dups_gpu.bam --in2 mark_dups_cpu.bam --noCigar --isize --flag --mate --mapQual
```

The output of this comparison should result in no differences.

Comparing BQSR Reports

The files generated by Parabricks and GATK4 should be exactly the same. There should be no output from the following command:

```
$ diff -w recal_gpu.txt recal_cpu.txt
```

Comparing VCF Files

To compare VCF files, use the GATK Concordance tools to get sensitivity and specificity of SNPs and INDELs. When the following command is run, variant accuracy results will be stored in `out.txt`.

```
$ gatk Concordance \ --evaluation result_gpu.vcf \ --truth result_cpu.vcf \ --summary out.txt
```

End User License Agreements

By pulling and using the Parabricks container, you accept the terms and conditions of the NVIDIA AI Product Agreement license.

Register for a free evaluation license to try NVIDIA AI Enterprise on your compatible, on-premises system or on the cloud!