Sentieon Documentation

Release 201808.01

Sentieon, Inc

Sep 30, 2018

Sentieon Manual

1

Introduction

1.1 Description

Sentieon Genomics software is a set of software tools that perform analysis of genomic data obtained from DNA sequencing.

1.2 Benefits and Value

The Sentieon Genomics software enables rapid and accurate analysis of next-generation sequence data. The Sentieon DNAseq pipelines enable germline variant calling on hundreds of thousands of samples simultaneously. The Sentieon TNseq pipelines enable accurate calling of somatic variants in paired tumor-normal samples or in an unpaired tumor samples. The Sentieon Genomics software produces more accurate results than other tools on third-party benchmarks with results obtained in one tenth the time of comparable pipelines.

1.3 Platform Requirements

Sentieon Genomics software is designed to run on Linux and other POSIX-compatible platforms.

For Linux systems, we recommended using the following Linux distributions or higher: RedHat/CentOS 6.5, Debian 7.7, OpenSUSE-13.2, or Ubuntu-14.04. Sentieon Genomics software has not been tested on older releases, and may not work.

For Apple OSX systems, OSX 10.9 (Mavericks) or higher is recommended.

Sentieon Genomics software requires at least 16GB of memory, although we recommend using 64 GB of memory. The alignment step is typically the most memory consuming stage; you can check section Section 8.7.3 for ways to control the memory used in alignment.

Sentieon Genomics software processes file data at speeds around 20-30MB/s per core using a state of the art server. In order to take advantage of the maximum speed that the software can provide, we recommend that you use high-speed SSD hard drives in your system, preferably two identical drives in a high speed RAID 0 striped volume configuration. When using a RAID 0 configuration, it is advised to use the drives to only store intermediate files, and move out any final results or important information out of those hard drives; this way, your server should have additional storage space to store the results.

Sentieon Genomics software requires that your system have python2.6.x or python2.7.x. You can check the version of the default python in your system by running:

```
python --version
```

If python2.6.x or python2.7.x is not the default python version in your system, you will need to install it and set an environmental variable to tell the software where the python2.6.x or python2.7.x binary is located

```
export SENTIEON_PYTHON=Python_2_X_binary_location
```

1.4 Installation procedure for a Linux Based system

In order to install the Sentieon Genomics software in a Linux based system, you need to follow these installation instructions:

1. Obtain the software release package. The software release package is named sentieon-genomics-201808.01.tar.gz, where 201808.01 is the release date.
2. Obtain the license file from Sentieon. The license file has extension .lic. The Sentieon license control is based on a lightweight floating license server process running on a node, and serving licenses though TCP. In a HPC cluster, the license server process is typically running in a special non-computing node on the cluster periphery that has unrestricted access to the outside world through HTTPS, and serves the licenses to the rest of the nodes in the cluster by listening to a specific TCP port that needs to be open within the cluster. The license server needs to have external https access to validate the license, while the computing client nodes do not need to have access to the internet. You need to provide Sentieon with the hostname (FQDN or IP) and port where you plan on deploying the license server.
3. Upload the software release package and license file to your server.
4. Decompress the release package with the command below. This will create a folder named sentieon-genomics-201808.01.

   `tar xvzf sentieon-genomics-201808.01.tar.gz`
5. Copy the license file to the directory that stores the license files (`LICENSE_DIR`). If you are running a license server process, start the license server.

   ```
   <SENTIEON_FOLDER>/bin/sentieon licsrvr --start LICENSE_DIR/LICENSE_FILE.lic
   ```
6. Set an environment variable in your system to indicate where the license is located (either `LICSRVR_HOST:LICSRVR_PORT` if you are using a license server or `LICENSE_DIR/LICENSE_FILE.lic` if you are using a license file)

   We recommend that you add this line to the shell scripts you use to run the Sentieon software, or to your shell profile:

   ```
   #if you are using a license server
   export SENTIEON_LICENSE=LICSRVR_HOST:LICSRVR_PORT
   #if you are using a license file
   export SENTIEON_LICENSE=LICENSE_DIR/LICENSE_FILE.lic
   ```

1.5 Installation procedure for an Apple OSX system

In order to install the Sentieon Genomics software in a Mac OSX based system, you need to follow these installation instructions:

1. Obtain the software release package. The software release package is named mac-sentieon-genomics-201808.01.tar.gz, where 201808.01 is the release date.
2. Obtain the license file. The license file has extension .lic.
3. Upload the software release package and license file to your computer.
4. Decompress the release package with the command below. This will create a folder named mac-sentieon-genomics-201808.01.

   ```
   tar xvzf mac-sentieon-genomics-201808.01.tar.gz
   ```

5. Copy the license file to the directory that stores the license files (`LICENSE_DIR`).
6. Set the environment variable in your system to indicate where the license is located (`LICENSE_DIR/LICENSE_FILE.lic`). We recommend that you add these lines to you shell profile `~/.bash_profile`:

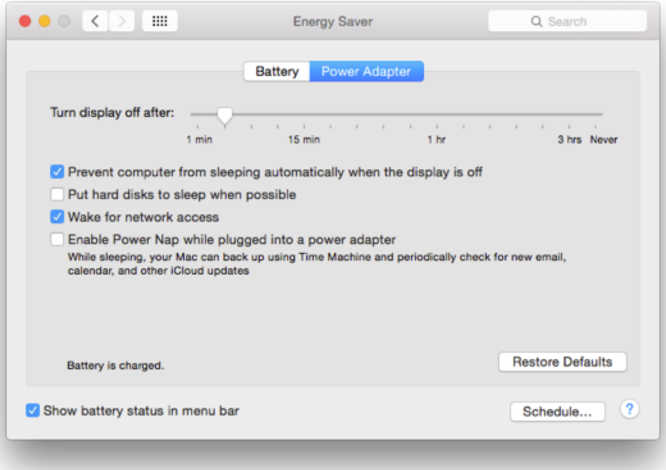
   ```
   export SENTIEON_LICENSE=LICENSE_DIR/LICENSE_FILE.lic
   ```

1.5.1 Using the software on an Apple OSX system

When using Sentieon Genomics software on a laptop using Apple OSX we recommend that you disable the energy saving capabilities of your computer, or temporarily change them while running commands.

To disable the energy saving capabilities of your computer:

1. Open the system preferences.
2. Select Energy Saver.
3. Make sure you check the option to “Prevent computer from sleeping automatically when display is off”, as shown in Fig. 1.1.



☑ Prevent computer from sleeping automatically when the display is off

□ Put hard disks to sleep when possible

☑ Wake for network access

□ Enable Power Nap while plugged into a power adapter
While sleeping, your Mac can back up using Time Machine and periodically check for new email, calendar, and other iCloud updates

Battery is charged.

☑ Show battery status in menu bar

Fig. 1.1: Setting the energy saving option in OSX

Alternatively, you can temporarily disable the energy saving capabilities of your computer by running the command from a Terminal:

```
pmset noidle
```

1.6 Keywords

Bioinformatics, DNA sequencing, Genomics.

2

Latest release notes and usage changes

2.1 Updates on this release

Type	Description
Feature	Improved performance of DNAscope and Machine learning model
Feature	Improved speed of Insert Size plotting when the sample has very large insert sizes
Bug-fix	Solved issue in GVCFtyper that could create a non conforming VCF when doing joint calling of DNAscope results.
Bug-fix	Solved issue in DNAModelApply that would cause an error when the input file does not contain a ModelID.
Bug-fix	Solved issue in Haplotyper when using the –given option that would generate a VCF without the LowQual filter definition in the header.
Bug-fix	Solved issue in tnhapfilter tool to include tumor sample in the header of the VCF

2.2 Usage changes from previous release

This released introduced the following changes in interface and usage:

- The binaries licsrvr and licclnt require using the bin/sentieon wrapper to be expecuted.

3 Typical usage for DNA seq

One of the typical uses of Sentieon Genomics software is to perform the bioinformatics pipeline for DNA analysis recommended in the Broad institute best practices described in https://www.broadinstitute.org/gatk/guide/best-practices. Fig. 3.1 illustrates such a typical bioinformatics pipeline.

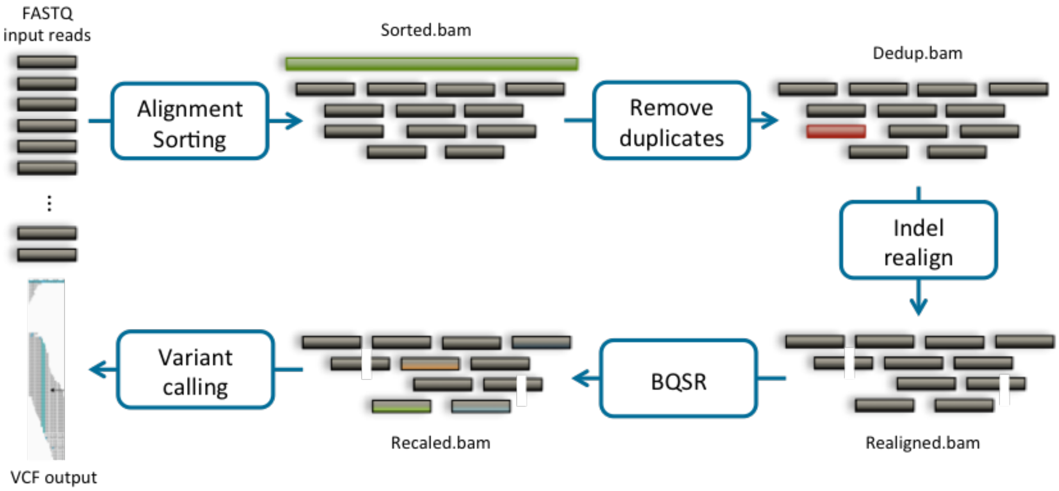


Fig. 3.1: Recommended bioinformatics pipeline for DNA variant calling analysis

3.1 General

In this bioinformatics pipeline you will need the following inputs:

- The FASTA file containing the nucleotide sequence of the reference genome corresponding to the sample you will analyze. The reference data needs to be pre-processed such that the data specified in Table 3.1 is available to the software. You can refer to *Preparing reference file for use* for instructions on how to generate the required files.

Table 3.1: Data requirements for the reference nucleotide sequence

Data	Description
.dict	Dictionary file
.fasta	Reference sequence file
.fasta.amb	BWA index file
.fasta.ann	BWA index file
.fasta.bwt	BWA index file
.fasta.fai	Index file
.fasta.pac	BWA index file
.fasta.sa	BWA index file

- One or multiple FASTQ files containing the nucleotide sequence of the sample to be analyzed. These files contain the raw reads from the DNA sequencing. The software supports inputting FASTQ files compressed using GZIP. The software only supports files containing quality scores in Sanger format (Phred+33).
- (Optional) The Single Nucleotide Polymorphism database (dbSNP) data that you want to include in the pipeline. The data is used in the form of a VCF file; you can use a VCF file compressed with bgzip and indexed.
- (Optional) Multiple collections of known sites that you want to include in the pipeline. The data is used in the form of a VCF file; you can use a VCF file compressed with bgzip and indexed.

The following steps compose the typical bioinformatics pipeline:

1. Map reads to reference: This step aligns the reads contained in the FASTQ files to map to a reference genome contained in the FASTA file. This step ensures that the data can be placed in context.
2. Calculate data metrics: This step produces a statistical summary of the data quality and the pipeline data analysis quality.
3. Remove duplicates: This step detects reads indicative that the same DNA molecules were sequenced several times. These duplicates are not informative and should not be counted as additional evidence.
4. (optional) Indel realignment: This step performs a local realignment around indels. This step is necessary as reads mapped on the edges of indels often get mapped with mismatching bases that are mapping artifacts. However, when using haplotype based callers such as Haplotyper or DNAscope, this step is not required, as the variant caller performs a local reassembly that provides most of the accuracy increase that results from Indel Realignment.
5. Base quality score recalibration (BQSR): This step modifies the quality scores assigned to individual read bases of the sequence read data. This action removes experimental biases caused by the sequencing methodology.
6. Variant calling: This step identifies the sites where your data displays variation relative to the reference genome, and calculates genotypes for each sample at that site.

3.2 Step by step usage for DNA seq

3.2.1 Map reads to reference

A single command is run to efficiently perform the alignment using BWA, and the creation of the BAM file and the sorting using Sentieon software:

```
(sentieon bwa mem -M -R '@RG\tID:GROUP_NAME\tSM:SAMPLE_NAME\tPL:PLATFORM' \
  -t NUMBER_THREADS REFERENCE SAMPLE [SAMPLE2] || echo -n 'error' ) \
  | sentieon util sort -r REFERENCE -o SORTED_BAM -t NUMBER_THREADS --sam2bam -i -
```

Inputs and options for BWA are described in its manual[1].

Alternatively, you can use other aligners that produce a file following the SAM format in stdout and replace the BWA portion of the command.

The following inputs are required for the command:

- **GROUP_NAME**: Readgroup identifier that will be added to the readgroup header line. The `RG:ID` needs to be unique among all the datasets that you plan on using, which is important when working with multiple input files as described in Section 9.3, or when performing a Tumor-Normal analysis as described in Section 6.
- **SAMPLE_NAME**: name of the sample that will be added to the readgroup header line.
- **PLATFORM**: name of the sequencing platform used to sequence the DNA. Possible options are ILLUMINA when the fastq files have been produced in an Illumina^{TM} machine; IONTORRENT when the fastq files have been produced in a Life Technologies^{TM} Ion-Torrent^{TM}machine.
- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system. We recommend that you use the same number of threads for both BWA and for the util binary.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that all additional reference data specified in Table 3.1 is available in the same location with consistent naming.
- **SAMPLE**: the location of the sample FASTQ file. If the data comes from pair ended sequencing technology, you will also need to input **SAMPLE2** as the corresponding pair sample FASTQ file.
- **SORTED_BAM**: the location and filename of the sorted mapped BAM output file. A corresponding index file (.bai) will be created.

BWA will produce slightly different results depending on the number of threads used in the command. This is due to the fact that BWA computes the insert size distribution on a chunk, whose size is dependent on the number of threads. To guarantee that the results are independent of the number of threads used, you should fix the chunk size in bases using option `-K 10000000`.

3.2.2 Calculate data metrics

A single command is run to generate 5 statistical summaries of the data quality and the pipeline data analysis quality results:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i SORTED_BAM \
  --algo GCBias --summary GC_SUMMARY_TXT GC_METRIC_TXT \
  --algo MeanQualityByCycle MQ_METRIC_TXT \
  --algo QualDistribution QD_METRIC_TXT \
  --algo InsertSizeMetricAlgo IS_METRIC_TXT  \
  --algo AlignmentStat ALN_METRIC_TXT
```

Four commands are run to generate the plots from the statistical summaries:

```
sentieon plot GCBias -o GC_METRIC_PDF GC_METRIC_TXT
sentieon plot MeanQualityByCycle -o MQ_METRIC_PDF MQ_METRIC_TXT
```

(continues on next page)

[1] http://bio-bwa.sourceforge.net/bwa.shtml

(continued from previous page)

```
sentieon plot QualDistribution -o QD_METRIC_PDF QD_METRIC_TXT
sentieon plot InsertSizeMetricAlgo -o IS_METRIC_PDF IS_METRIC_TXT
```

The following inputs are required for the commands:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **SORTED_BAM**: the location where the previous mapping stage stored the result.
- **GC_SUMMARY_TXT**: the location and filename of the GC bias metrics summary results output file.
- **GC_METRIC_TXT**: the location and filename of the GC bias metrics results output file.
- **MQ_METRIC_TXT**: the location and filename of the mapping quality metrics results output file.
- **QD_METRIC_TXT**: the location and filename of the quality/depth metrics results output file.
- **IS_METRIC_TXT**: the location and filename of the insertion size metrics results output file.
- **ALN_METRIC_TXT**: the location and filename of the alignment metrics results output file.
- **GC_METRIC_PDF**: the location and filename of the GC bias metrics report output file.
- **MQ_METRIC_PDF**: the location and filename of the mapping quality metrics report output file.
- **QD_METRIC_PDF**: the location and filename of the quality/depth metrics report output file.
- **IS_METRIC_PDF**: the location and filename of the insertion size metrics report output file.

3.2.3 Remove duplicates

Two individual commands are run to remove duplicates on the BAM file after alignment and sorting. The first command collects read information, and the second command performs the deduping.

```
sentieon driver -t NUMBER_THREADS -i SORTED_BAM \
  --algo LocusCollector --fun score_info SCORE_TXT
sentieon driver -t NUMBER_THREADS -i SORTED_BAM \
  --algo Dedup --rmdup --score_info SCORE_TXT  \
  --metrics DEDUP_METRIC_TXT DEDUP_BAM
```

The following inputs are required for the commands:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **SORTED_BAM**: the location where the previous mapping stage stored the result.
- **SCORE_TXT**: the location and filename of the temporary score output file. Make sure that the same file is used for both commands.
- **DEDUP_METRICS_TXT**: the location and filename of the dedup metrics results output file.
- **DEDUP_BAM**: the location and filename of the deduped BAM output file. A corresponding index file (.bai) will be created.

3.2.4 Indel realignment (optional)

A single command is run to perform local realignment around indels on the BAM file after alignment, sorting and deduping.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE \
  -i DEDUP_BAM --algo Realigner [-k KNOWN_SITES] REALIGNED_BAM
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **DEDUP_BAM**: the location where the previous deduping stage stored the result.
- **REALIGNED_BAM**: the location and filename of the realigned BAM output file. A corresponding index file (.bai) will be created.

The following inputs are optional for the command:

- **KNOWN_SITES**: the location of the VCF file used as a set of known sites. You can include multiple collections of known sites by repeating the `-k KNOWN_SITES option`.

3.2.5 Base quality score recalibration (BQSR)

A single command is run to calculate the required modification of the quality scores assigned to individual read bases of the sequence read data; the actual recalibration is applied during the variant calling stage. The input BAM file to this command depends on whether the Indel Realignment step was performed.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE \
  -i REALIGNED_BAM/DEDUP_BAM --algo QualCal [-k KNOWN_SITES] RECAL_DATA.TABLE
```

Three commands are run to apply the recalibration and create a report on the base quality sore recalibration. The first command applies the recalibration to calculate the post calibration data table and additionally apply the recalibration on the BAM file, the second one creates the data for plotting; the third command plots the calibration data tables, both pre and post, into graphs in a pdf.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i REALIGNED_BAM/DEDUP_BAM \
  -q RECAL_DATA.TABLE --algo QualCal [-k KNOWN_SITES] \
  RECAL_DATA.TABLE.POST [--algo ReadWriter RECALIBRATED_BAM]
sentieon driver -t NUMBER_THREADS --algo QualCal --plot \
  --before RECAL_DATA.TABLE --after RECAL_DATA.TABLE.POST RECAL_RESULT.CSV
sentieon plot QualCal -o BQSR_PDF RECAL_RESULT.CSV
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **REALIGNED_BAM**: the location where the previous realignment stage stored the result.
- **DEDUP_BAM**: the location where the previous deduping stage stored the result.
- **RECAL_DATA.TABLE**: the location and filename of the recalibration table.

- **RECAL_DATA.TABLE.POST**: the location and filename of the temporary post recalibration table
- **RECAL_RESULT.CSV**: the location and filename of the temporary recalibration results output file used for plotting.
- **BQSR_PDF**: the location and filename of the BSQR results output file.

The following inputs are optional for the command:

- **KNOWN_SITES**: the location of the VCF file used as a set of known sites. You can include multiple collections of known sites by repeating the `-k KNOWN_SITES` option.
- **RECALIBRATED_BAM**: the location and filename of the recalibrated BAM output file. A corresponding index file (.bai) will be created. This output is optional as Sentieon variant callers can perform the recalibration on the fly using the before recalibration BAM plus the recalibration table

3.2.6 Variant calling

A single command is run to call variants and additionally apply the BQSR calculated before. The input BAM file to this command depends on whether the Indel Realignment step was performed.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i REALIGNED_BAM/DEDUP_BAM \
  -q RECAL_DATA.TABLE --algo Haplotyper [-d dbSNP] VARIANT_VCF
```

You may want to rerun only the variant calling, for example to use Unified Genotyper variant calling algorithm. In that case, you do not need to re-apply the BQSR, and can use the previously recalibrated BAM:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i RECALIBRATED_BAM \
  --algo Genotyper [-d dbSNP] VARIANT_VCF
```

In both cases, using the recalibrated BAM or the BAM before recalibration plus the recalibration data table will give identical results; however, you should be careful not to use the recalibration data table together with the already recalibrated BAM, as that would apply the recalibration twice, leading to incorrect results.

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **REALIGNED_BAM**: the location where the previous realignment stage stored the result.
- **DEDUP_BAM**: the location where the previous deduping stage stored the result.
- **RECAL_DATA.TABLE**: the location where the previous BQSR stage stored the result.
- **RECALIBRATED_BAM**: the location of the recalibrated BAM file.
- **VARIANT_VCF**: the location and filename of the variant calling output file. A corresponding index file (.idx) will be created. The tool will output a compressed file by using .gz extension.

The following inputs are optional for the command:

- **dbSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP) that will be used to label known variants. You can only use one dbSNP file.

4 Typical usage for DNA scope

The Sentieon Genomics software includes an improved algorithm to perform the variant calling step of germline DNA analysis. The pipeline used for DNAscope is similar to the DNAseq one described in *Typical usage for DNA seq*. DNAscope can perform structural variant calling in addition to calling SNPs and small indels.

4.1 General

In this bioinformatics pipeline you will need the following inputs:

- The FASTA file containing the nucleotide sequence of the reference genome corresponding to the sample you will analyze.
- One or multiple FASTQ files containing the nucleotide sequence of the sample to be analyzed. These files contain the raw reads from the DNA sequencing. The software supports inputting FASTQ files compressed using GZIP. The software only supports files containing quality scores in Sanger format (Phred+33).
- (Optional) The Single Nucleotide Polymorphism database (dbSNP) data that you want to include in the pipeline. The data is used in the form of a VCF file; you can use a VCF file compressed with bgzip and indexed.
- (Optional) Multiple collections of known sites that you want to include in the pipeline. The data is used in the form of a VCF file; you can use a VCF file compressed with bgzip and indexed.
- (For SNPs and small indels calling) A machine learning model file.

The following steps compose the typical bioinformatics pipeline for a tumor-normal matched pair:

1. Pre-process the sample using a DNA seq pipeline like the one introduced in Section 3, with the following stages:
   (a) Map reads to reference.
   (b) Calculate data metrics.
   (c) Remove duplicates.
   (d) (optional) Indel realignment.
   (e) Base quality score recalibration (BQSR).

2. Variant calling using DNAscope: This step identifies the sites where your data displays variation relative to the reference genome, and calculates genotypes for each sample at that site.

4.2 Step by step usage

For the mapping, duplicate removal, indel realignment and base quality score recalibration stages, please refer to Section 3.2 for detailed usage instructions.

4.2.1 Germline variant calling

A single command is run to call variants and additionally apply the BQSR calculated before. The input BAM file to this command depends on whether the Indel Realignment step was performed.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i REALIGNED_BAM/DEDUP_BAM \
  -q RECAL_DATA.TABLE --algo DNAscope [-d dnSNP] VARIANT_VCF
```

Alternatively, you can use a previously recalibrated BAM:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i RECALIBRATED_BAM \
    --algo DNAscope [-d dnSNP] VARIANT_VCF
```

In both cases, using the recalibrated BAM or the BAM before recalibration plus the recalibration data table will give identical results; however, you should be careful not to use the recalibration data table together with the already recalibrated BAM, as that would apply the recalibration twice, leading to incorrect results.

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **REALIGNED_BAM**: the location where the previous realignment stage stored the result.
- **RECAL_DATA.TABLE**: the location where the previous BQSR stage stored the result.
- **RECALIBRATED_BAM**: the location of the recalibrated BAM file.
- **VARIANT_VCF**: the location and filename of the variant calling output file. A corresponding index file (.idx) will be created. The tool will output a compressed file by using .gz extension.

The following inputs are optional for the command:

- **dbSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP) that will be used to label known variants. You can only use one dbSNP file.

4.2.2 Germline variant calling with a machine learning model

Two individual commands are run to call variants while additionally applying the BQSR calculated before, and to apply the machine learning model. The input BAM file to this command depends on whether the Indel Realignment step was performed.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i REALIGNED_BAM/DEDUP_BAM \
  --algo DNAscope [-d dnSNP] --model ML_MODEL TMP_VARIANT_VCF
sentieon driver -t NUMBER_THREADS -r REFERENCE --algo DNAModelApply \
  --model ML_MODEL -v TMP_VARIANT_VCF VARIANT_VCF
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **REALIGNED_BAM**: the location where the previous realignment stage stored the result.
- **TMP_VARIANT_VCF**: the location and filename of the variant calling output of DNAscope. This is a temporary file.
- **VARIANT_VCF**: the location and filename of the variant calling output. A corresponding index file (.idx) will be created. The tool will output a compressed file by using .gz extension.
- **ML_MODEL**: the location of the machine learning model file. In the DNAscope command the model will be used to determine the settings used in variant calling.

The following inputs are optional for the command:

- **dbSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP) that will be used to label known variants. You can only use one dbSNP file.

4.2.3 Structural variant calling

In order to perform structural variant calling you need to add the option to output break-end (BND) information to the DNAscope command; this is done by enabling the bnd (break-end) variant type. Two individual commands are run to perform structural variant calling while additionally applying the BQSR calculated before. The input BAM file to the first command depends on whether the Indel Realignment step was performed.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i REALIGNED_BAM/DEDUP_BAM \
  -q RECAL_DATA.TABLE --algo DNAscope --var_type bnd \
  [-d dbSNP] TMP_VARIANT_VCF
sentieon driver -t NUMBER_THREADS -r REFERENCE --algo SVSolver  \
  -v TMP_VARIANT_VCF STRUCTURAL_VARIANT_VCF
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **TMP_VARIANT_VCF**: the location and filename of the variant calling output file from DNAscope, which includes the BND information. This is a temporary file when calling structural variants.
- **STRUCTURAL_VARIANT_VCF**: the location and filename of the variant calling output file containing the structural variants. A corresponding index file (.idx) will be created. The tool will output a compressed file by using .gz extension.

5 Typical usage for TN seq

Another of the typical uses of Sentieon Genomics software is to perform the bioinformatics pipeline for Tumor-Normal analysis recommended in the Broad institute in https://www.broadinstitute.org/gatk/guide/presentations?id=6007 - bpp2B[2]. Fig. 5.1 illustrates such a typical bioinformatics pipeline.

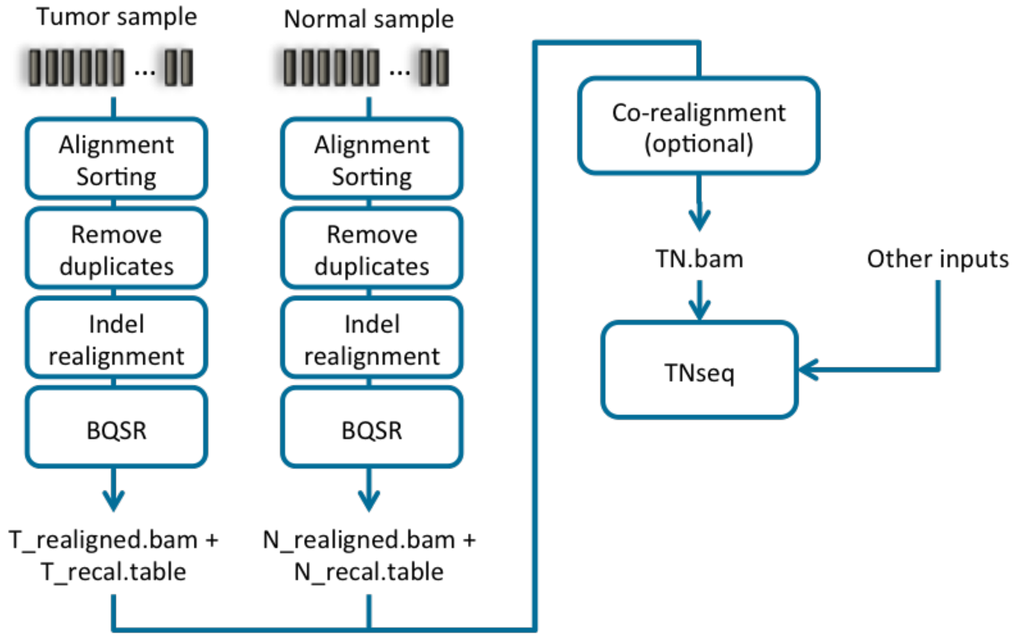


Fig. 5.1: Recommended bioinformatics pipeline for Tumor-Normal analysis

5.1 General

In this bioinformatics pipeline you will need the following inputs:

[2] https://www.broadinstitute.org/gatk/guide/presentations?id=6007#bpp2B

- The FASTA file containing the nucleotide sequence of the reference genome corresponding to the sample you will analyze.
- Two sets of FASTQ files containing the nucleotide sequence of the sample to be analyzed, one for the tumor sample and one for the matched normal sample. These files contain the raw reads from the DNA sequencing. The software supports inputting FASTQ files compressed using GZIP. The software only supports files containing quality scores in Sanger format (Phred+33).
- (Optional) The Single Nucleotide Polymorphism database (dbSNP) data that you want to include in the pipeline. The data is used in the form of a VCF file.
- (Optional) Multiple collections of known sites that you want to include in the pipeline. The data is used in the form of a VCF file.

You can also include in the pipeline the following optional inputs that will help the algorithms detect artifacts and remove false positives:

- Panel of normal VCF: list of common errors that appear as variants from multiple unrelated normal samples. The contents of this file will be used to identify variants that are more likely to be germline variants, and filter them as such.
- Cosmic VCF: data from the Catalogue of Somatic Mutations in Cancer (COSMIC) representing a list of known tumor related variants. The contents of this file will be used to reduce the germline risk factor of the variants. You need to use the same COSMIC file as the one used to generate the Panel of Normal VCF.

If you do not have accessed to a normal (non tumor) sample matched to the tumor sample, the Panel of Normal VCF and the Cosmic VCF inputs are highly recommended.

The following steps compose the typical bioinformatics pipeline for a tumor-normal matched pair:

1. Independently pre-process the Tumor and Normal samples using a DNA seq pipeline like the one introduced in Section 3, with the following stages:
    (a) Map reads to reference; you need to make sure that the SM sample tag is different between the tumor and the normal samples, as you will need it as an argument in the somatic variant calling.
    (b) Calculate data metrics.
    (c) Remove duplicates.
    (d) Indel realignment.
    (e) Base quality score recalibration (BQSR).
2. Indel co-realignment: This step performs a local realignment around indels for the combined data of both tumor and normal sample.
3. Somatic variant calling on the co-realigned BAM or on the two individual bam: This step identifies the sites where the cancer genome data displays somatic variations relative to the normal genome, and calculates genotypes at that site.

The following steps compose the typical bioinformatics pipeline for a tumor sample without normal matched sample (Fig. 5.2):

1. Pre-process the Tumor sample using a DNA seq pipeline like the one introduced in Section 3, with the following stages:
    (a) Map reads to reference.
    (b) Calculate data metrics.
    (c) Remove duplicates.
    (d) Indel realignment.

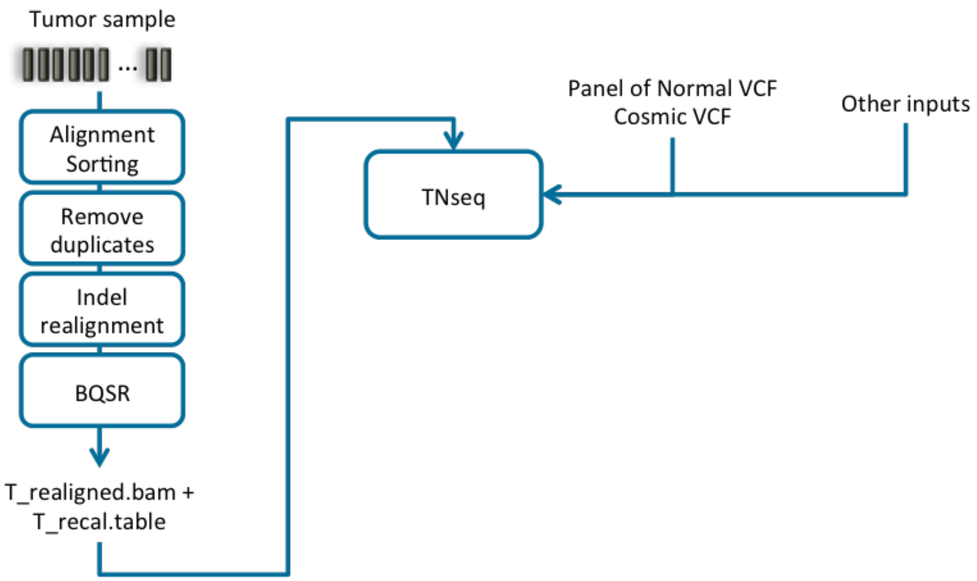


Fig. 5.2: Recommended bioinformatics pipeline for Tumor only analysis

   (e) Base quality score recalibration (BQSR).

   (f) (optional) Apply recalibration and create the recalibrated BAM file.

2. Variant calling on the tumor sample using information from a generic panel of normal samples: This step identifies the sites where the cancer genome data displays somatic variations relative to the panel of normal genome, and calculates genotypes at that site.

5.2 Step by step usage

For the mapping, duplicate removal, indel realignment and base quality score recalibration stages, please refer to Section 3.2 for detailed usage instructions.

5.2.1 Indel co-realignment

A single command is run to perform local realignment around indels on the combination of both Tumor and Normal BAM files after pre-processing.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_REALIGN_BAM \
  -q TUMOR_RECAL_DATA.TABLE -i NORMAL_REALIGN_BAM -q NORMAL_RECAL_DATA.TABLE \
  --algo Realigner [-k KNOWN_SITES] COREALIGNED_BAM
```

Alternatively, you can use already recalibrated BAM files to achieve the same results.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_RECALIBRATED_BAM \
  -i NORMAL_RECALIBRATED_BAM --algo Realigner [-k KNOWN_SITES] COREALIGNED_BAM
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.

- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **TUMOR_REALIGNED_BAM**: the location of the pre-processed BAM file after individual realignment for the TUMOR sample.
- **TUMOR_RECAL_DATA.TABLE**: the location where the BQSR stage for the TUMOR sample stored the result.
- **NORMAL_REALIGNED_BAM**: the location of the pre-processed BAM file after individual realignment for the NORMAL sample.
- **NORMAL_RECAL_DATA.TABLE**: the location where the BQSR stage for the NORMAL sample stored the result.
- **TUMOR_RECALIBRATED_BAM**: the location of the pre-processed BAM file after recalibration for the TUMOR sample.
- **NORMAL_RECALIBRATED_BAM**: the location of the pre-processed BAM file after recalibration for the NORMAL sample.
- **COREALIGNED_BAM**: the location and filename of the realigned BAM output file. A corresponding index file (.bai) will be created.

The following inputs are optional for the command:

- **KNOWN_SITES**: the location of the VCF file used as a set of known sites. You can include multiple collections of known sites by repeating the `-k KNOWN_SITES` option.

5.2.2 Variant discovery with matched normal sample

A single command is run to call variants on the tumor-normal matched pair.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i COREALIGNED_BAM \
  --algo TNhaplotyper --tumor_sample TUMOR_SAMPLE_NAME \
  --normal_sample NORMAL_SAMPLE_NAME [--dbsnp DBSNP] OUT_TN_VCF
```

You can also use the TNhaplotyper2 algorithm to call variants, in which case two commands are used:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i COREALIGNED_BAM \
  --algo TNhaplotyper2 --tumor_sample TUMOR_SAMPLE_NAME \
  --normal_sample NORMAL_SAMPLE_NAME [--dbsnp DBSNP] TMP_OUT_TN_VCF
sentieon tnhapfilter --tumor_sample TUMOR_SAMPLE_NAME \
   --normal_sample NORMAL_SAMPLE_NAME -v TMP_OUT_TN_VCF OUT_TN_VCF
```

Alternatively, you can use TNsnv to call SNVs only:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i COREALIGNED_BAM \
  --algo TNsnv --tumor_sample TUMOR_SAMPLE_NAME \
  --normal_sample NORMAL_SAMPLE_NAME [--dbsnp DBSNP] \
  [--call_stats_out OUT_CALL_STATS] OUT_TN_VCF
```

Alternatively, if you did not run co-realignment, you can use the before co-realignment BAM files to perform the somatic calling:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_REALIGN_BAM \
  -q TUMOR_RECAL_DATA.TABLE -i NORMAL_REALIGN_BAM \
  -q NORMAL_RECAL_DATA.TABLE --algo TNhaplotyper \
```

(continues on next page)

(continued from previous page)

```
  --tumor_sample TUMOR_SAMPLE_NAME --normal_sample NORMAL_SAMPLE_NAME \
  [--dbsnp DBSNP] OUT_TN_VCF
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **COREALIGNED_BAM**: the location and filename of the co-realigned BAM file from co-realignment stage.
- **TUMOR_SAMPLE_NAME**: sample name used for tumor sample in Map reads to reference stage.
- **NORMAL_SAMPLE_NAME**: sample name used for normal sample in Map reads to reference stage.
- **TMP_OUT_TN_VCF**: the location and file name of the output file from TNhaplotyper2; this is a temporary file.
- **OUT_TN_VCF**: the location and file name of the output file containing the variants.

The following inputs are optional for the command:

- **DBSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP). The variants in the dbSNP will be more likely to be marked as germline as they require more evidence of absence in the normal. You can only use one dbSNP file.
- **OUT_CALL_STATS**: the location and file name of the output file containing the call stats of the discovered variants. The file is in text format. This argument is only needed when using TNsnv.

5.2.3 Step by step usage when missing a matched normal sample

When missing a matched normal sample, you will need to replace the matched normal sample with a generic panel of normal file generated with the cosmic database file.

A single command is run to call variants on the tumor only sample.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_REALIGNED_BAM \
  -q TUMOR_RECAL_DATA.TABLE --algo TNhaplotyper \
  --tumor_sample TUMOR_SAMPLE_NAME --pon PANEL_OF_NORMAL_VCF \
  --cosmic COSMIC_DB_VCF [--dbsnp DBSNP] OUT_TN_VCF
```

Alternatively, you can use already recalibrated BAM files to achieve the same results.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_RECALIBRATED_BAM \
  --algo TNhaplotyper --tumor_sample TUMOR_SAMPLE_NAME \
  --pon PANEL_OF_NORMAL_VCF --cosmic COSMIC_DB_VCF \
  [--dbsnp DBSNP] OUT_TN_VCF
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.

- **TUMOR_REALIGNED_BAM**: the location of the pre-processed BAM file after realignment for the TUMOR sample.
- **TUMOR_RECAL_DATA.TABLE**: the location where the BQSR stage for the TUMOR sample stored the result.
- **TUMOR_RECALIBRATED_BAM**: the location of the pre-processed BAM file after recalibration for the TUMOR sample.
- **TUMOR_SAMPLE_NAME**: sample name used for tumor sample in Map reads to reference stage.
- **PANEL_OF_NORMAL_VCF**: the location and name of panel of normal VCF file.
- **COSMIC_VCF**: the location and name of the cosmic database VCF file.
- **OUT_TN_VCF**: the location and file name of the output file containing the variants.

The following inputs are optional for the command:

- **DBSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP). The variants in the dbSNP will be more likely to be marked as germline as they require more evidence of absence in the normal. You can only use one dbSNP file.

5.2.4 Generating a Panel of Normal VCF file

In order to generate your own Panel of Normal VCF file, you will need to run the following command on every normal sample you want to use in the panel:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i NORMAL_RECALIBRATED_BAM \
  --algo TNhaplotyper --detect_pon \
  --cosmic COSMIC_VCF [--dbsnp DBSNP] OUT_NORMAL_VCF
```

We recommend that you create the panel of normal file with the corresponding algorithm that you plan to use for the somatic mutation calling. If you plan to use TNsnv for the calling, you should use TNsnv for the generation of the panel of normal.

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **NORMAL_RECALIBRATED_BAM**: the location of the pre-processed BAM file for the normal sample after Sentieon Readwrite stage.
- **COSMIC_VCF**: the location and name of the cosmic database VCF file.
- **OUT_NORMAL_VCF**: the location and name of the output VCF file containing the relevant variants for the input normal sample.

The following inputs are optional for the command:

- **DBSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP) that will be used to label known variants. You can only use one dbSNP file.

After you have generated all VCF files you want to include in the panel, you need to merge them into a single Panel of Normal VCF. You can use bcftools[3] for that purpose:

[3] https://samtools.github.io/bcftools/bcftools.html

```
BCF=/path_to_bcftools
export BCFTOOLS_PLUGINS=$BCF/plugins
DIR=/path_to_normal_vcf_file
$BCF/bcftools merge -m any -f PASS,. --force-samples $DIR/*.vcf.gz |\
$BCF/bcftools plugin fill-AN-AC |\
$BCF/bcftools filter -i 'SUM(AC)>1' > panel_of_normal.vcf
```

6 Typical usage for TN scope

A use of Sentieon Genomics software is to perform the bioinformatics pipeline for Tumor only or Tumor-Normal analysis using the new TNscope algorithms for somatic variant and structural variant detection. Fig. 6.1 illustrates such a typical bioinformatics pipeline.

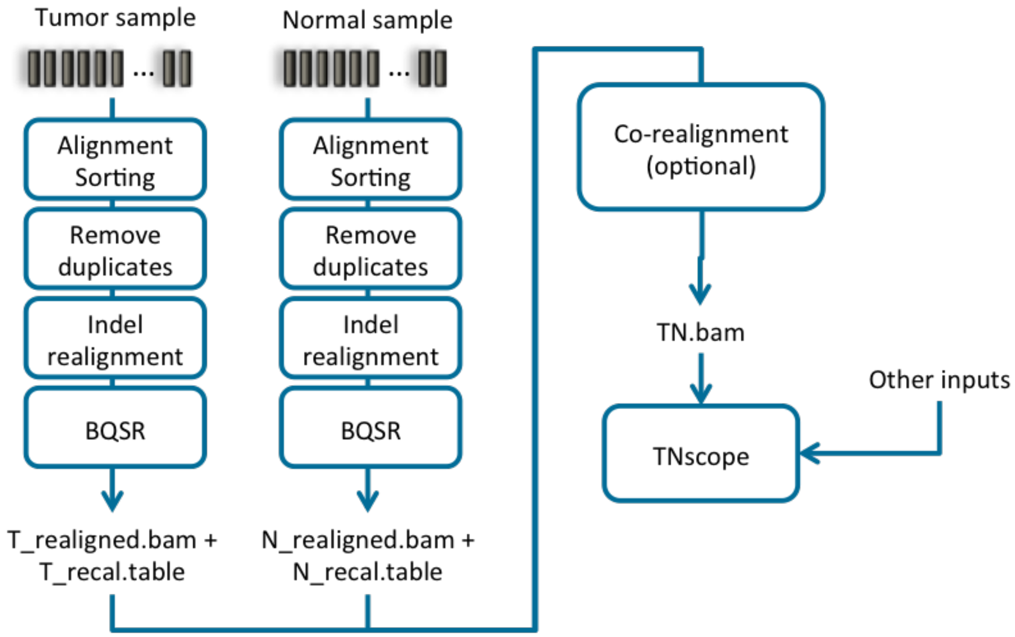


Fig. 6.1: Recommended pipeline for TNscope based Tumor-Normal analysis

6.1 General

In this bioinformatics pipeline you will need the following inputs:

- The FASTA file containing the nucleotide sequence of the reference genome corresponding to the sample you will analyze.
- Two sets of FASTQ files containing the nucleotide sequence of the sample to be analyzed, one for the tumor sample and one for the matched normal sample. These files contain the raw reads from the DNA

sequencing. The software supports inputting FASTQ files compressed using GZIP. The software only supports files containing quality scores in Sanger format (Phred+33).

- (Optional) The Single Nucleotide Polymorphism database (dbSNP) data that you want to include in the pipeline. The data is used in the form of a VCF file.
- (Optional) Multiple collections of known sites that you want to include in the pipeline. The data is used in the form of a VCF file.

The following steps compose the typical bioinformatics pipeline for a tumor-normal matched pair:

1. Independently pre-process the Tumor and Normal samples using a DNA seq pipeline like the one introduced in Section 3, with the following stages:
   (a) Map reads to reference; you need to make sure that the SM sample tag is different between the tumor and the normal samples, as you will need it as an argument in the somatic variant calling. You also need to make sure that the RG:ID for both samples is different and unique.

   (b) Calculate data metrics.

   (c) Remove duplicates.

   (d) (optional) Indel realignment.

   (e) Base quality score recalibration (BQSR).
2. (optional) Indel co-realignment: This step performs a local realignment around indels for the combined data of both tumor and normal sample.
3. Somatic variant calling on the co-realigned sample: This step identifies the sites where the cancer genome data displays somatic variations relative to the normal genome, and calculates genotypes at that site.

6.2 Step by step usage

For the mapping, duplicate removal, indel realignment and base quality score recalibration stages, please refer to Section 3.2 for detailed usage instructions.

6.2.1 (optional) Indel co-realignment

A single command is run to perform local realignment around indels on the combination of both Tumor and Normal BAM files after pre-processing.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_REALIGN_BAM \
  -q TUMOR_RECAL_DATA.TABLE -i NORMAL_REALIGN_BAM \
  -q NORMAL_RECAL_DATA.TABLE --algo Realigner [-k KNOWN_SITES] COREALIGNED_BAM
```

Alternatively, you can use already recalibrated BAM files to achieve the same results.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i TUMOR_RECALIBRATED_BAM \
  -i NORMAL_RECALIBRATED_BAM --algo Realigner [-k KNOWN_SITES] COREALIGNED_BAM
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.

- **TUMOR_REALIGNED_BAM**: the location of the pre-processed BAM file after individual realignment for the TUMOR sample.
- **TUMOR_RECAL_DATA.TABLE**: the location where the BQSR stage for the TUMOR sample stored the result.
- **NORMAL_REALIGNED_BAM**: the location of the pre-processed BAM file after individual realignment for the NORMAL sample.
- **NORMAL_RECAL_DATA.TABLE**: the location where the BQSR stage for the NORMAL sample stored the result.
- **TUMOR_RECALIBRATED_BAM**: the location of the pre-processed BAM file after recalibration for the TUMOR sample.
- **NORMAL_RECALIBRATED_BAM**: the location of the pre-processed BAM file after recalibration for the NORMAL sample.
- **COREALIGNED_BAM**: the location and filename of the realigned BAM output file. A corresponding index file (.bai) will be created.

The following inputs are optional for the command:

- **KNOWN_SITES**: the location of the VCF file used as a set of known sites. You can include multiple collections of known sites by repeating the `-k KNOWN_SITES` option.

6.2.2 Variant discovery with or without matched normal sample

A single command is run to call variants on the tumor-normal matched pair.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i COREALIGNED_BAM \
 --algo TNscope --tumor_sample TUMOR_SAMPLE_NAME \
 --normal_sample NORMAL_SAMPLE_NAME [--dbsnp DBSNP] OUT_TN_VCF
```

If you skipped the optional Indel co-realignment, the command would use the individual tumor and normal BAM files with the corresponding recalibration tables:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE \
  -i TUMOR_REALIGN_BAM/TUMOR_DEDUP_BAM -q TUMOR_RECAL_DATA.TABLE \
  -i NORMAL_REALIGN_BAM/NORMAL_DEDUP_BAM -q NORMAL_RECAL_DATA.TABLE \
  --algo TNscope \
  --tumor_sample TUMOR_SAMPLE_NAME --normal_sample NORMAL_SAMPLE_NAME \
  [--dbsnp DBSNP] OUT_TN_VCF
```

If you do not have a matched normal sample, you can skip the normal sample BAM file and sample name inputs, as those are not required; however, due to the absence of normal sample, germline variants will be present in the output file:

```
sentieon driver -t NUMBER_THREADS -r REFERENCE \
  -i TUMOR_REALIGN_BAM/TUMOR_DEDUP_BAM -q TUMOR_RECAL_DATA.TABLE \
  --algo TNscope --tumor_sample TUMOR_SAMPLE_NAME \
  [--dbsnp DBSNP] OUT_TN_VCF
```

In order to filter out germline variants when missing a matched normal sample, you can replace the matched normal sample with a generic panel of normal file generated with the cosmic database file using TNhaplotyper as described in Section 5.2.4.

A single command is run to call variants on the tumor only sample.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE \
  -i TUMOR_REALIGNED_BAM/TUMOR_DEDUP_BAM -q TUMOR_RECAL_DATA.TABLE \
  --algo TNscope --tumor_sample TUMOR_SAMPLE_NAME \
  --pon PANEL_OF_NORMAL_VCF --cosmic COSMIC_DB_VCF [--dbsnp DBSNP] OUT_TN_VCF
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **COREALIGNED_BAM**: the location and filename of the co-realigned BAM file from co-realignment stage.
- **TUMOR_SAMPLE_NAME**: sample name used for tumor sample in Map reads to reference stage.
- **OUT_TN_VCF**: the location and file name of the output file containing the variants.

The following inputs are optional for the command:

- **NORMAL_SAMPLE_NAME**: sample name used for normal sample in Map reads to reference stage.
- **DBSNP**: the location of the Single Nucleotide Polymorphism database (dbSNP). The variants in the dbSNP will be more likely to be marked as germline as they require more evidence of absence in the normal. You can only use one dbSNP file.
- **PANEL_OF_NORMAL_VCF**: the location and name of panel of normal VCF file.
- **COSMIC_VCF**: the location and name of the cosmic database VCF file.

7 Typical usage for RNA variant calling

One of the typical uses of Sentieon Genomics software is to perform the bioinformatics pipeline for RNA analysis recommended in the Broad institute best practices described in http://gatkforums.broadinstitute.org/wdl/discussion/3892/the-gatk-best-practices-for-variant-calling-on-rnaseq-in-full-detail.

7.1 General

In this bioinformatics pipeline you will need the following inputs:

- The FASTA file containing the nucleotide sequence of the reference genome corresponding to the sample you will analyze.
- A BAM file containing the information for the sample to be analyzed. These files contain the reads from the RNA sequencing aligned using STAR.
- (Optional) The Single Nucleotide Polymorphism database (dbSNP) data that you want to include in the pipeline. The data is used in the form of a VCF file; you can use a VCF file compressed with bgzip and indexed.
- (Optional) Multiple collections of known sites that you want to include in the pipeline. The data is used in the form of a VCF file; you can use a VCF file compressed with bgzip and indexed.

The following steps compose the typical bioinformatics pipeline:

1. Remove duplicates: This step detects reads indicative that the same RNA molecules were sequenced several times. These duplicates are not informative and should not be counted as additional evidence.
2. Split reads at junction: This step splits the RNA reads into exon segments by getting rid of Ns while maintaining grouping information, and hard-clips any sequences overhanging into the intron regions. Additionally, the step will reassign the mapping qualities from STAR to be consistent with what is expected in subsequent steps by converting from quality 255 to 60.
3. Indel realignment: This step performs a local realignment around indels. This step is necessary as reads mapped on the edges of indels often get mapped with mismatching bases that are mapping artifacts.
4. Base quality score recalibration (BQSR): This step modifies the quality scores assigned to individual read bases of the sequence read data. This action removes experimental biases caused by the sequencing methodology.
5. Variant calling: This step identifies the sites where your data displays variation relative to the reference genome, and calculates genotypes for each sample at that site.

7.2 Step by step usage

The Remove duplicates stage, Indel realignment stage and Base quality score recalibration (BQSR) stages have identical usage to the DNA seq stages; for these stages, you can refer to Section 3.2 for detailed usage instructions.

7.2.1 Split reads at junction

A single command is run to perform the splitting of reads into exon segments and reassigning the mapping qualities from STAR.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i DEDUP_BAM \
  --algo RNASplitReadsAtJunction --reassign_mapq 255:60 SPLIT_BAM
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **DEDUP_BAM**: the location where the previous deduping stage stored the result.
- **SPLIT_BAM**: the location and filename of the BAM output file containing the split reads. A corresponding index file (.bai) will be created.

7.2.2 RNA Variant calling

A single command is run to call variants and additionally apply the BQSR calculated before. The RNA variant calling should be done using the Haplotyper algorithm; additionally you should use the option `--trim_soft_clip` and a lower minimum phred-scaled confidence threshold than for DNAseq variant calling; you should set `call_conf` to 20 and `emit_conf` to 20 instead of the default of 30.

```
sentieon driver -t NUMBER_THREADS -r REFERENCE -i REALIGNED_BAM \
  -q RECAL_DATA.TABLE --algo Haplotyper --trim_soft_clip  \
  --call_conf 20 --emit_conf 20 [-d dbSNP] VARIANT_VCF
```

The following inputs are required for the command:

- **NUMBER_THREADS**: the number of computer threads that will be used in the calculation. We recommend that the number does not exceed the number of computing cores available in your system.
- **REFERENCE**: the location of the reference FASTA file. You should make sure that the reference is the same as the one used in the mapping stage.
- **REALIGNED_BAM**: the location where the previous realignment stage stored the result.
- **RECAL_DATA.TABLE**: the location where the previous BQSR stage stored the result.
- **VARIANT_VCF**: the location and filename of the variant calling output file. A corresponding index file (.idx) will be created. The tool will output a compressed file by using .gz extension.

The following inputs are optional for the command:

- **dbSNP:** the location of the Single Nucleotide Polymorphism database (dbSNP) that will be used to label known variants. You can only use one dbSNP file.

8

Detailed usage of the tools

8.1 DRIVER binary

DRIVER is the binary used to execute all stages of the bioinformatics pipeline. A single call to the driver binary can run multiple algorithms; for example, the metrics stage is implemented as a single command call to driver running multiple algorithms.

8.1.1 DRIVER syntax

The general syntax of the DRIVER binary is:

```
sentieon driver OPTIONS --algo ALGORITHM ALGO_OPTION OUTPUT \
[--algo ALGORITHM2 ALGO_OPTION2 OUTPUT2]
```

In the case of running multiple algorithms in the same driver call, the OPTIONS are shared among all the algorithms.

The arguments (OPTIONS) for this command include:

- `-t NUMBER_THREADS`: number of computing threads that will be used by the software to run parallel processes. The argument is optional; if omitted, the driver tool will use as many threads as the server has.
- `-r REFERENCE`: location of the reference FASTA file. This argument is required in all algorithms except LocusCollector and Dedup.
- `-i INPUT_FILE`: location of the BAM input file. This argument can be used multiple times to have the software use multiple input files and the result would be the same as if the input BAM files are merged into one BAM file. This argument is required in all algorithms except for `CollectVCMetrics`, `GVCFtyper`, `DNAModelApply`, `SVSolver`, `TNModelApply`, `TNModelTrain`, `VarCal` and `ApplyVarCal`.
- `-q QUALITY_RECALIBRATION_TABLE`: location of the quality recalibration table output from the BQSR stage that will be used as an input during the BQSR stage and Variant Calling stage. This argument can be used multiple times to have the software use multiple input calibration tables; each calibration table will apply a recalibration to BAM files by matching the readgroup of the reads
- `--interval INTERVAL`: interval in the reference that will be used in the software. Only a single input `INTERVAL` will be considered: if you repeat the `--interval` option, only the `INTERVAL` in the last one will be considered. `INTERVAL` can be specified as:

  - CONTIG[:START-END]: calculation will be done only on the corresponding contig. START and END are optional numbers to further reduce the interval; both are first-base-1 based. You can input a comma-separated list of multiple contigs.
  - BED_FILE: location of the BED file containing the intervals.
  - PICARD_INTERVAL_FILE: location of the file containing the intervals, following the Picard interval standard.
- --interval_padding PADDING_SIZE: adds PADDING_SIZE bases padding to the edges of the input intervals. The default is 0.
- --help: option to display help. The option can be used together with the --algo ALGORITHM to display help for the specific algorithm.
- --read_filter FILTER,OPTION=VALUE,OPTION=VALUE: perform a filter or transformation of reads prior to the application of the algorithm. Please refer to Section 8.1.3 to get additional information on the available filters and their functionality.
- --temp_dir DIRECTORY: determines where the temporary files will be stored. The default is the folder where the command is run ($PWD).
- --skip_no_coor: determines whether to skip unmapped reads.
- --cram_read_options decode_md=0: CRAM input option to turn off the NM/MD tag in the input CRAM.
- --replace_rg ORIG_RG="NEW_RG_STRING": modifies the @RG Read group tag of the next BAM input file to update the ORIG_RG with the new information; the NEW_RG_STRING needs to be a valid and complete string. This argument can be used multiple times, and each will affect the next -i BAM input. You can check Section 9.9 for a detail example on its usage

The supported algorithms (ALGORITHM) for this command are:

- LocusCollector, Dedup: used in the remove duplicates stage.
- Realigner: used in the indel realignment stage.
- QualCal: used in the base quality recalibration stage.
- MeanQualityByCycle, QualDistribution, GCBias, AlignmentStat, InsertSizeMetricAlgo, HsMetricAlgo, CoverageMetrics, BaseDistributionByCycle, QualityYield, WgsMetricsAlgo, SequenceArtifactMetricsAlgo: used to calculate QC metrics.
- Genotyper, Haplotyper: used for germline variant calling analysis.
- DNAscope and DNAModelApply: used in DNAscope germline variant calling analysis.
- DNAscope and SVsolver: used in the DNAscope germline structural variant analysis.
- ReadWriter: used to output the BAM file after base quality recalibration or merge multiple BAM files.
- VarCal, ApplyVarCal: used in the VQSR stage.
- GVCFtyper: used for germline joint variant calling analysis.
- TNsnv, TNhaplotyper, TNhaplotyper2: used for somatic tumor-normal or tumor only analysis.
- TNscope, TNModelApply: used for somatic tumor-normal somatic and structural variant analysis.
- RNASplitReadsAtJunction: used in the stage to split RNA reads at junctions.
- ContaminationAssessment: used to identify cross-sample contamination from BAM files.
- CollectVCMetrics: used to calculate post-variant calling metrics.

8.1.2 DRIVER ALGORITHM syntax

8.1.2.1 LocusCollector ALGORITHM

The LocusCollector algorithm collects read information that will be used for removing duplicate reads.

The input to the LocusCollector algorithm is a BAM file; its output is the score file indicating which reads are likely duplicates.

The LocusCollector algorithm requires the following ALGO_OPTION:

- `--fun SCORE`: scoring function to use. Possible values for SCORE are:
  - score_info: calculates the score of a read pair for the Dedup algorithm. This is the default score function.

8.1.2.2 Dedup ALGORITHM

The Dedup algorithm performs the removing of duplicate reads.

The input to the Dedup algorithm is a BAM file; its output is the BAM file after removing duplicate reads.

The Dedup algorithm requires the following ALGO_OPTION:

- `--score_info LOCUS_COLLECTOR_OUTPUT`: location of the output file of the LocusCollector command call.

The Dedup algorithm accepts the following optional ALGO_OPTION:

- `--rmdup`: set this option to remove duplicated reads in the output BAM file. If this option is not set, the duplicated read will be marks as such with a flag, but not removed from the BAM file.
- `--cram_write_options compressor=[gzip|bzip2|lzma|rans]`: CRAM output compression options. compressor=gzip is default if not defined.
- `--cram_write_options version=[2.1|3.0]`: CRAM output version options. version=2.1 is default if not defined.
- `--metrics METRICS_FILE`: location and filename of the output file containing the metrics data from the deduping stage.
- `--optical_dup_pix_dist DISTANCE`: determine the maximum distance between two duplicate reads for them to be considered optical duplicates. The default is 100.
- `--bam_compression COMPRESSION_LEVEL[0-9]`: gzip compression level for the output BAM file. The default value is 6.
- `--output_dup_read_name`: when using this option, the output of the command will not be a BAM file with marked/removed duplicates, but a list of read names for reads that were marked as duplicate.
- `--dup_read_name DUPLICATE_READ_NAME_FILE`: when using this input all reads contained in the DUPLICATE_READ_NAME_FILE will be marked as duplicate, regardless of whether they are primary or non-primary.

8.1.2.3 Realigner ALGORITHM

The Realigner algorithm performs the indel realignment.

The input to the Realigner algorithm is a BAM file; its output is the BAM file after realignment.

The Realigner algorithm accepts the following optional ALGO_OPTION:

- `-k KNOWN_SITES`: location of the VCF file used as a set of known sites. The known sites will be used to help identify likely sites where the realignment is necessary; only indel variants in the file will be used. You can include multiple collections of known sites by specifying multiple files and repeating the `-k KNOWN_SITES` option.
- `--interval_list INTERVAL`: interval in the reference that will be used in the calculation of the realign targets. INTERVAL can be specified as:
  - BED_FILE: location of the BED file containing the intervals.
  - PICARD_INTERVAL_FILE: location of the file containing the intervals, following the Picard interval standard.
- `--bam_compression COMPRESSION_LEVEL[0-9]`: gzip compression level for the output BAM file. The default value is 6.
- `--cram_write_options compressor=[gzip|bzip2|lzma|rans]`: CRAM output compression options. compressor=gzip is default if not defined.
- `--cram_write_options version=[2.1|3.0]`: CRAM output version options. version=2.1 is default if not defined.

8.1.2.4 QualCal ALGORITHM

The QualCal algorithm calculates the recalibration table necessary to do the BQSR. The QualCal algorithm also applies the recalibration to calculate the data required to create a report, and creates the data required to create a report.

The recalibration math depends on platform (PL) tag of the ReadGroup; the QualCal algorithm supports the following platforms: ILLUMINA, ION_TORRENT, LS454, PACBIO, COMPLETE_GENOMICS, Support for sequencing data from the SOLID platform is not currently implemented.

The input to the QualCal algorithm is a BAM file; its output is a recalibration table or the csv file containing the data required to create a report.

The QualCal algorithm accepts the following optional ALGO_OPTION:

- `-k KNOWN_SITES`: location of the VCF file used as a set of known sites. The known sites will be used to make sure that known locations do not get artificially low quality scores by misidentifying true variants errors in the sequencing methodology. You can include multiple collections of known sites by specifying multiple files and repeating the `-k KNOWN_SITES` option. We strongly recommend using as many known sites as possible, as otherwise the recalibration will consider variant sites to be sequencing errors.
- `--plot`: indicates whether the command is being used to generate the data required to create a report.
- `--cycle_val_max`: maximum allowed cycle value for the cycle covariate.
- `--before RECAL_TABLE`: location of the previously calculated recalibration table; it will be used to apply the recalibration.
- `--after RECAL_TABLE.POST`: location of the previously calculated results of applying the recalibration table; it will be used to calculate the data required to create a report.

8.1.2.5 MeanQualityByCycle ALGORITHM

The MeanQualityByCycle algorithm calculates the mean base quality score for each sequencing cycle.

The input to the MeanQualityByCycle algorithm is a BAM file; its output is the metrics data.

The MeanQualityByCycle algorithm does not accept any ALGO_OPTION.

8.1.2.6 QualDistribution ALGORITHM

The QualDistribution algorithm calculates the number of bases with a specific base quality score.

The input to the QualDistribution algorithm is a BAM file; its output is the metrics data.

The QualDistribution algorithm does not accept any ALGO_OPTION.

8.1.2.7 GCBias ALGORITHM

The GCBias algorithm calculates the GC bias in the reference and the sample.

The input to the GCBias algorithm is a BAM file; its output is the metrics data.

The GCBias algorithm accepts the following optional ALGO_OPTION:

- `--summary SUMMARY_FILE`: location and filename of the output file summarizing the GC Bias metrics.
- `--accum_level LEVEL`: determines the accumulation levels. The possible values of LEVEL are ALL_READS, SAMPLE, LIBRARY, READ_GROUP. The default is ALL_READS.
- `--also_ignore_duplicates`: determines whether the output metrics will be calculated using unique non duplicated reads.

8.1.2.8 HsMetricAlgo ALGORITHM

The HsMetricAlgo algorithm calculates the Hybrid Selection specific metrics for the sample and the AT/GC dropout metrics for the reference.

The input to the HsMetricAlgo algorithm is a BAM file; its output is the metrics data.

The HsMetricAlgo algorithm requires the following ALGO_OPTION:

- `--targets_list TARGETS_FILE`: location and filename of the interval list input file that contains the locations of the targets.
- `--baits_list TARGETS_FILE`: location and filename of the interval list input file that contains the locations of the baits used.
- `--clip_overlapping_reads`: determines whether to clip overlapping reads during the calculation.
- `--min_map_qual QUALITY`: determines the filtering quality of the reads used. Any reads with mapping quality less than QUALITY will be filtered out.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored.
- `--coverage_cap COVERAGE`: determines the maximum coverage limit used in the histogram.

8.1.2.9 AlignmentStat ALGORITHM

The AlignmentStat algorithm calculates statistics about the alignment of the reads.

The input to the AlignmentStat algorithm is a BAM file; its output is the metrics data.

The AlignmentStat algorithm accepts the following optional ALGO_OPTION:

- `--adapter_seq SEQUENCE_LIST`: the sequence of the adapters used in the sequencing, provided as a comma separated list. The default value is the list of default Illumina adapters.

8.1.2.10 InsertSizeMetricAlgo ALGORITHM

The InsertSizeMetricAlgo algorithm calculates the statistical distribution of insert sizes.

The input to the InsertSizeMetricAlgo algorithm is a BAM file; its output is the metrics data.

The InsertSizeMetricAlgo algorithm does not accept any ALGO_OPTION.

8.1.2.11 CoverageMetrics ALGORITHM

The CoverageMetrics algorithm calculates the depth coverage of the BAM file. The coverage is aggregated by interval if the `--interval` option is included at the driver level; if no `--interval` option is included, the aggregation per interval will be done per contig for all bases in the reference and the per locus coverage output file will be about 60GB. The coverage is aggregated by gene if a RefSeq file is included.

The input to the CoverageMetrics algorithm is a BAM file; its outputs are files containing the metrics data organized by partition, aggregation and output. Possible outputs are:

- summary: contains the depth data.
- statistics: contains the histogram of loci with specific depth.
- cumulative_coverage_counts: contains the histogram of loci with depth larger than x.
- cumulative_coverage_proportions: contains the normalized histogram of loci with depth larger than x.

Examples of output files when the output name is OUTPUT:

- OUTPUT: the per locus coverage with no partition.
- OUTPUT.sample_summary: the summary for PARTITION_GROUP sample, aggregated over all bases.
- OUTPUT. library_interval_statistics: the statistics for PARTITION_GROUP library, aggregated by interval.

The CoverageMetrics algorithm accepts the following optional ALGO_OPTION:

- `--partition PARTITION_GROUP`: determine how to partition the data. Possible values are readgroup or a comma separated combination of the RG attributes, namely sample, platform, library, center. The default value is “sample”. You can include multiple partition groups by repeating the `-partition` option, and each ouput file will be created once per `--partition` option.
- `--gene_list REFSEQ_FILE`: location of the RefSeq file used to aggregate the results of the CoverageMetrics algorithm to the gene level.
- Filtering options:
  - `--min_map_qual` and `--max_map_qual MAP_QUALITY`: determines the filtering quality of the reads used. Any reads with mapping quality less than QUALITY will be filtered out.
  - `--min_base_qual` and `--max_base_qual  QUALITY`: determines the filtering quality of the bases used. Any base with quality less than QUALITY will be ignored.
  - `--cov_thresh THRESHOLD`: add percentage of bases in the aggregation that have coverage larger than the threshold. You can include multiple thresholds by repeating the `--cov_thresh` argument.
- Omit output options:
  - `--omit_base_output`: skip the output of the per locus coverage with no partition. This option can be used when you do not use intervals to save space.
  - `--omit_sample_stat`: skip the output of summary results aggregated over all bases (_summary)

  - `--omit_locus_stat`: skip the output of all histogram files (both _cumulative_coverage_counts and _cumulative_coverage_proportions).
  - `--omit_interval_stat`: skip the output of all interval statistics files (_interval_statistics).
- `--count_type TYPE`: determines how to deal with overlapping reads from the same fragment. Possible options are:
  - 0: to count overlapping reads even if they come from the same fragment. This is the default value.
  - 1: to count overlapping reads
  - 2: to count overlapping reads only if the reads in the fragment have consistent bases.
- `--print_base_counts`: include the number of "AGCTND" in the output per locus coverage with no partition.
- `--include_ref_N`: include the coverage data in loci where the reference genome is set to N.
- `--ignore_del_sites`: ignore the coverage data in loci where there are deletions.
- `--include_del`: this argument will interact with others as follows:
  - if ignore_del_sites is off, count Deletion as depth
  - if print_base_counts is on, include number of 'D'
- `--histogram_scale [log/linear] --histogram_low MIN_DEPTH`, `--histogram_high MAX_DEPTH`, `--histogram_bin_count NUM_BINS`: determine the scale type, bin and sizes for histograms. The default values are log, 1, 500, 499.

8.1.2.12 CollectVCMetrics ALGORITHM

The CollectVCMetrics algorithm collects metrics related to the variants present in the input VCF.

The input to the CollectVCMetrics algorithm is a VCF file and a DBSNP file; its output is a pair of files containing information about the variants from the VCF file.

The CollectVCMetrics algorithm requires the following ALGO_OPTION:

- `-d dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP). Only one file is supported.
- `-v INPUT`: location of the VCF file on which the metrics will be calculated. Only one file is supported.

8.1.2.13 BaseDistributionByCycle

The BaseDistributionByCycle algorithm calculates the nucleotide distribution per sequencer cycle.

The input to the BaseDistributionByCycle algorithm is a BAM file; its output is the metrics data.

The BaseDistributionByCycle algorithm accepts the following optional ALGO_OPTION:

- `--aligned_reads_only`: determines whether to calculate the base distribution over aligned reads only.
- `--pf_reads_only`: determines whether to calculate the base distribution over PF reads only.

8.1.2.14 QualityYield

The QualityYield algorithm collects metrics related to reads that pass quality thresholds and Illumina-specific filters.

The input to the QualityYield algorithm is a BAM file; its output is the metrics data.

The QualityYield algorithm accepts the following optional ALGO_OPTION:

- `--include_secondary`: determines whether to include bases from secondary alignments in the calculation.
- `--include_supplementary`: determines whether to include bases from supplementary alignments in the calculation.

8.1.2.15 WgsMetricsAlgo

The WgsMetricsAlgo algorithm collects metrics related to the coverage and performance of whole genome sequencing (WGS) experiments.

The input to the WgsMetricsAlgo algorithm is a BAM file; its output is the metrics data.

The WgsMetricsAlgo algorithm accepts the following optional ALGO_OPTION:

- `--min_map_qual QUALITY`: determines the filtering quality of the reads used in the calculation. Any read with quality less than QUALITY will be ignored. The default value is 20.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in the calculation. Any base with quality less than QUALITY will be ignored. The default value is 20.
- `--coverage_cap COVERAGE`: determines the maximum coverage limit for the histogram. Any position with coverage higher than COVERAGE will have its coverage set to COVERAGE.
- `--sample_size SIZE`: determines the Sample Size used for the Theoretical Het Sensitivity sampling. The default value is 10000.
- `--include_unpaired`: determines whether to count unpaired reads and paired reads with one end unmapped.
- `--base_qual_histogram`: determines whether to report the base quality histogram.

8.1.2.16 SequenceArtifactMetricsAlgo

The SequenceArtifactMetricsAlgo algorithm collects metrics that quantify single-base sequencing artifacts and OxoG artifacts.

The input to the SequenceArtifactMetricsAlgo algorithm is a BAM file; its output is the metrics data.

The SequenceArtifactMetricsAlgo algorithm accepts the following optional ALGO_OPTION:

- `--dbsnp FILE`: location of the Single Nucleotide Polymorphism database (dbSNP) used to exclude regions around known polymorphisms. Only one file is supported.
- `--min_map_qual QUALITY`: determines the filtering quality of the reads used in the calculation. Any read with quality less than QUALITY will be ignored. The default value is 30.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in the calculation. Any base with quality less than QUALITY will be ignored. The default value is 20.
- `--include_unpaired`: determines whether to count unpaired reads and paired reads with one end unmapped.

- `--include_duplicates`: determines whether to count duplicated reads.
- `--include_non_pf_reads`: determines whether to count non-PF reads.
- `--min_insert_size ISIZE`: determines the filtering insert size of the reads used in the calculation. Any read with insert size less than ISIZE will be ignored. The default value is 60.
- `--max_insert_size ISIZE`: determines the filtering insert size of the reads used in the calculation. Any read with insert size larger than ISIZE will be ignored. The default value is 600.
- `--tandem_reads`: determines whether the mate pairs are being sequenced from the same strand.
- `--context_size SIZE`: determined the number of context bases to include on each side. The default value is 1.

8.1.2.17 Genotyper ALGORITHM

The Genotyper algorithm performs the Unified Genotyper variant calling.

The input to the Genotyper algorithm is a BAM file; its output is a VCF file.

The Genotyper algorithm accepts the following optional ALGO_OPTION:

- `--annotation 'ANNOTATION_LIST'`: determines additional annotations that will be added to the output VCF. Use a comma separated list to enable or disable annotations. Include 'none' to remove the default annotations; prefix annotations with the exclamation point (!) to disable the specific annotation. See Section 9.4 for supported annotations.
- `-d dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP) used to label known variants. Only one file is supported.
- `--var_type VARIANT_TYPE`: determine which variant types will be called; possible values for VARIANT_TYPE are:
  - SNP to call only Single Nucleotide Polymorphism. This is the default behavior.
  - INDEL to call only insertion-deletions.
  - both to call both SNPs and INDELs.
- `--call_conf CONFIDENCE`: determine the threshold of variant quality to call a variant. Variants with quality less than CONFIDENCE will be removed.
- `--emit_conf CONFIDENCE`: determine the threshold of variant quality to emit a variant. Variants with quality less than CONFIDENCE will be not be added to the output VCF file.
- `--emit_mode MODE`: determines what calls will be emitted. Possible values for MODE are:
  - variant: emit calls only at confident variant sites. This is the default behavior.
  - confident: emit calls at confident variant sites or confident reference sites.
  - all: emit all calls, regardless of their confidence.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 17.
- `--ploidy PLOIDY`: determines the ploidy number of the sample being processed. The default is 2.
- `--given GIVEN_VCF`: perform variant calling using only the variants provided in the GIVEN_VCF. The calling will only evaluate the locus and alleles provided in the file.

8.1.2.18 Haplotyper ALGORITHM

The Haplotyper algorithm performs the Haplotype variant calling.

The input to the Haplotyper algorithm is a BAM file; its output is a VCF file.

The Haplotyper algorithm accepts the following optional ALGO_OPTION:

- `--annotation 'ANNOTATION_LIST'`: determines additional annotations that will be added to the output VCF. Use a comma separated list to enable or disable annotations. Include 'none' to remove the default annotations; prefix annotations with the exclamation point (!) to disable the specific annotation. See Section 9.4 for supported annotations.
- `-d dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP) used to label known variants. Only one file is supported.
- `--call_conf CONFIDENCE`: determine the threshold of variant quality to call a variant. Variants with quality less than CONFIDENCE will be removed. This option is ignored when the `--emit_mode` is gvcf.
- `--emit_conf CONFIDENCE`: determine the threshold of variant quality to emit a variant. Variants with quality less than CONFIDENCE will be not be added to the output VCF file. This option is ignored when the `--emit_mode` is gvcf.
- `--emit_mode MODE`: determines what calls will be emitted. Possible values for mode are:
  - variant: emit calls only at confident variant sites. This is the default behavior.
  - confident: emit calls at confident variant sites or confident reference sites.
  - all: emit all calls, regardless of their confidence.
  - gvcf: emits additional information required for joint calling. This option is required if you want to perform joint calling using the GVCFtyper algorithm.
- `--gq_bands LIST_OF_BANDS`: determines the bands that will be used to compress variants of similar genotype quality (GQ) that will be emitted as a single VCF record in the GVCF output file. The LIST_OF_BANDS is a comma-separated list of bands where each band is defined by START-END/STEP. The default is 1-60,60-99/10,99.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 10.
- `--pcr_indel_model MODEL`: PCR indel model used to weed out false positive indels more or less aggressively. The possible MODELs are: NONE (used for PCR free samples), and HOSTILE, AGGRESSIVE and CONSERVATIVE, in order of decreasing aggressiveness. The default value is CONSERVATIVE.
- `--phasing [1/0]`: flag to enable or disable phasing in the output when using `emit_mode GVCF`. The default value is 1 (on) and this flag has no impact when using an emit_mode other than GVCF. Phasing is only calculated for diploid samples.
- `--ploidy PLOIDY`: determines the ploidy number of the sample being processed. The default is 2.
- `--prune_factor FACTOR`: minimum pruning factor in local assembly; paths with fewer supporting kmers than FACTOR will be pruned from the graph. The default value is 2.
- `--trim_soft_clip`: determines whether soft clipped bases in the reads should be included in the variant calling. This argument is only recommended to process RNA reads.
- `--given GIVEN_VCF`: perform variant calling using only the variants provided in the GIVEN_VCF. The calling will only evaluate the locus and alleles provided in the file. This option cannot be used in conjunction with `--emit_mode gvcf`.

- `--bam_output OUTPUT_BAM`: output a BAM file containing modified reads after the local reassembly done by the variant calling. This option should only be used in conjunction with a small bed file for troubleshooting purposes.
- `--snp_heterozygosity HETEROZIGOSITY`: determines the expected heterozygosity value used to compute prior likelihoods in SNP calling.
- `--indel_heterozygosity HETEROZIGOSITY`: determines the expected heterozygosity value used to compute prior likelihoods in INDEL calling.

8.1.2.19 ReadWriter ALGORITHM

The ReadWriter algorithm outputs the result of applying the Base Quality Score Recalibration to a file.

The ReadWriter algorithm can also merge BAM files, and/or convert them into cram files.

The input to the ReadWriter algorithm is one or multiple BAM files and one or multiple recalibration tables; its output is the BAM file after recalibration. If the output file extension is CRAM, a CRAM file will be created. If multiple input files were used, the output file will be the result of merging all the files.

The ReadWriter algorithm accepts the following optional ALGO_OPTION:

- `--bam_compression COMPRESSION_LEVEL[0-9]`: gzip compression level for the output BAM file. The default value is 6.
- `--cram_write_options compressor=[gzip|bzip2|lzma|rans]`: CRAM output compression options. compressor=gzip is default if not defined.
- `--cram_write_options version=[2.1|3.0]`: CRAM output version options. version=2.1 is default if not defined.
- `--read_flag_mask pass=[all|FLAG_LIST_pass],drop=[FLAG_LIST_drop]`: read filtering settings. FLAG_LIST is a '+' separated list of multiple Flag numbers or case insensitive Flag names. The default read_flag_mask is "pass=all,drop=0x0" as the ReadWriter algorithm allows all reads. As an example, to write a BAM file that does not contain either duplicated reads or unmapped reads, you can use `--read_flag_mask pass=all,drop=DUP+UNMAP`. The table below is a reminder of the flag definition and their naming convention.

Flag #	Flag name	Description
0x1	PAIRED	paired-end (or multiple-segment) sequencing technology
0x2	PROPER_PAIR	each segment properly aligned according to the aligner
0x4	UNMAP	segment unmapped
0x8	MUNMAP	next segment in the template unmapped
0x10	REVERSE	SEQ is reverse complemented
0x20	MREVERSE	SEQ of the next segment in the template is reversed
0x40	READ1	the first segment in the template
0x80	READ2	the last segment in the template
0x100	SECONDARY	secondary alignment
0x200	QCFAIL	not passing quality controls
0x400	DUP	PCR or optical duplicate
0x800	SUPPLEMENTARY	supplementary alignment

We recommend running the ReadWriter algorithm at the same command call as one of the variant calls, to reduce overhead.

8.1.2.20 GVCFtyper ALGORITHM

The GVCFtyper algorithm performs the joint variant calling of multiple samples, provided that each single sample has been previously processed using the Haplotyper algorithm with the option `--emit_mode gvcf`.

The GVCFtyper algorithm has no input in the driver level, its output is a VCF containing the joint called variants for all samples.

The GVCFtyper algorithm requires the following ALGO_OPTION:

- `-v INPUT`: location of the GVCF file from the variant calling algorithm performed on a single sample using the extra option `--emit_mode gvcf`. You can include GVCF files from multiple samples by specifying multiple files and repeating the `-v INPUT` option. You can use VCF files compressed with bgzip and indexed.
- Alternatively, you can input a list of GVCF files at the end of the command after the output file. Thus, the following 2 commands are interchangeable:

```
sentieon driver -r REFERENCE --algo GVCFtyper \
  -v s1_VARIANT_GVCF -v s2_VARIANT_GVCF -v s3_VARIANT_GVCF VARIANT_VCF
sentieon driver -r REFERENCE --algo GVCFtyper \
  VARIANT_VCF s1_VARIANT_GVCF s2_VARIANT_GVCF s3_VARIANT_GVCF
```

- You can read in the GVCF file list from a text file using the following commands:

```
gvcf_argument=""
while read -r line; do
 gvcf_argument=$gvcf_argument" -v $line"
done < "list_of_gvcfs"
sentieon driver -r REFERENCE --algo GVCFtyper $gvcf_argument output-joint.vcf
```

- You can also read in the GVCF file list from a text file using the following command leveraging the stdin pipe (-):

```
cat list_of_gvcfs | sentieon driver -r REFERENCE --algo GVCFtyper output-joint.vcf -
```

- You could input all files from a specific folder using the following command:

```
sentieon driver -r REFERENCE --algo GVCFtyper output-joint.vcf sample*.g.vcf
```

The GVCFtyper algorithm accepts the following optional ALGO_OPTION:

- `--annotation 'ANNOTATION_LIST'`: determines additional annotations that will be added to the output VCF. Use a comma separated list to enable or disable annotations. Include 'none' to remove the default annotations; prefix annotations with the exclamation point (!) to disable the specific annotation. See Section 9.4 for supported annotations.
- `-d dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP) used to label known variants. Only one file is supported.
- `--call_conf CONFIDENCE`: determine the threshold of variant quality to call a variant. Variants with quality less than CONFIDENCE will be removed.
- `--emit_conf CONFIDENCE`: determine the threshold of variant quality to emit a variant. Variants with quality less than CONFIDENCE will be not be added to the output VCF file.
- `--emit_mode MODE`: determines what calls will be emitted. Possible values for mode are:
  - variant: emit calls only at confident variant sites. This is the default behavior.
  - confident: emit calls at confident variant sites or confident reference sites.

  - all: emit all calls, regardless of their confidence.

- `--max_alt_alleles NUMBER`: Maximum number of alternate alleles. The default value is 100.
- `--snp_heterozygosity HETEROZIGOSITY`: determines the expected heterozygosity value used to compute prior likelihoods in SNP calling.
- `--indel_heterozygosity HETEROZIGOSITY`: determines the expected heterozygosity value used to compute prior likelihoods in INDEL calling.

8.1.2.21 VarCal ALGORITHM

The VarCal algorithm calculates the Variant Quality Score Recalibration (VQSR). VQSR assigns a well-calibrated probability score to individual variant calls, to enable more accurate control in determining the most likely variants. For that, VQSR uses highly confident known sites to build a recalibration model and determine the probability that called sites are true. For more information about the algorithm, you can check http://gatkforums.broadinstitute.org/discussion/39/variant-quality-score-recalibration-vqsr. For information on the recommended resources to use in VQSR, you can check https://www.broadinstitute.org/gatk/guide/article?id=1259.

The VarCal algorithm has no input in the driver level, its output is a recalibration file containing additional annotations related to the VQSR.

The VarCal algorithm requires the following ALGO_OPTION:

- `-v INPUT`: location of the VCF file from the variant calling algorithm; you can use a VCF file compressed with bgzip and indexed.
- `--tranches_file TRANCHES_FILE`: location and filename of the file containing the partition of the call sets into quality tranches.
- `--resource RESOURCE_FILE --resource_param PARAM`: location of the VCF file used as a training/truth resource in VQSR, followed by parameters determining how the file will be used. You can include multiple collections by specifying multiple files and repeating the `--resource dbSNP_FILE --resource_param PARAM` option. The PARAM argument follows the syntax:

```
LABEL,known=IS_KNOWN,training=IS_TRAIN,truth=IS_TRUTH,prior=PRIOR
```

  - **LABEL** is a descriptive name for the resource.
  - **IS_KNOWN** can be *true* or *false*, and determines whether the sites contained in the resource will be used to stratify output metrics.
  - **IS_TRAIN** can be *true* or *false*, and determines whether the sites contained in the resource will be used for training the recalibration model.
  - **IS_TRUTH** can be *true* or *false*, and determines whether the sites contained in the resource will be considered true sites.
  - **PRIOR** is a value that reflects your confidence in how reliable the resource is as a truth set.

The VarCal algorithm accepts the following optional ALGO_OPTION:

- `--srand RANDOM_SEED`: determines the seed to use in the random number generation. You can set RANDOM_SEED to 0 and the software will use the random seed from your computer. In order to generate a deterministic result, you should use a non-zero RANDOM_SEED and set the NUMBER_THREADS to 1.
- `--annotation ANNOTATION`: determine annotation that will be used during the recalibration. You can include multiple annotations in the optimization by repeating the `--annotation ANNOTATION` option. You can use all annotations present in the original variant call file.

- `--var_type VARIANT_TYPE`: determine which variant types will be recalibrated; possible values for VARIANT_TYPE are:
  - SNP to recalibrate only Single Nucleotide Polymorphism. This is the default behavior.
  - INDEL to recalibrate only insertion-deletions.
  - (do not use) BOTH to recalibrate both SNPs and INDELs. This setting SHOULD NOT be used, as VQSR should be performed independently for SNPs and INDELs.
- `--tranche TRANCH_THRESHOLD`: normalized quality threshold for each tranche; the TRANCH_THRESHOLD number is a number between 0 and 100. Multiple instances of the option are allowed that will create as many tranches as there are thresholds. The default values are 90, 99, 99.9 and 100.
- `--max_gaussians MAX_GAUSS`: determines the maximum number of Gaussians that will be used for the positive recalibration model. The default value is 8 for SNP and 4 for INDEL.
- `--max_neg_gaussians MAX_GAUSS`: determines the maximum number of Gaussians that will be used for the negative recalibration model. The default value is 2.
- `--max_iter MAX_ITERATIONS`: determines the maximum number of iterations for the Expectation Maximization (EM) optimization. The default value is 150.
- `--max_mq MAPQ`: indicates the maximum MQ in your data, which will be used to perform a logit jitter transform of the MQ to make the distribution closer to a Gaussian.
- `--aggregate_data AGREGATE_VCF`: location of an additional VCF file containing variants called from other similar samples; these additional data will increase the effective sample size for the statistical model calibration. Multiple instances of the option are allowed.
- `--plot_file PLOT_FILE`: location of the temporary file containing the necessary data to generate the reports from the VarCal algorithm.

8.1.2.22 ApplyVarCal ALGORITHM

The ApplyVarCal algorithm combines the output information from the VQSR with the original variant information.

The ApplyVarCal algorithm has no input in the driver level; its output is a copy of the original VCF containing additional annotations from the VQSR.

The ApplyVarCal algorithm requires the following ALGO_OPTION:

- `-v INPUT`: location of the VCF file from the variant calling algorithm. It should be the same as the one used in the VarCal algorithm; you can use a VCF file compressed with bgzip and indexed.
- `--recal VARIANT_RECAL_DATA`: location of the VCF file output from the VarCal algorithm.
- `--tranches_file TRANCHES_FILE`: location of the tranches file output from the VarCal algorithm.
- `--var_type VARIANT_TYPE`: determine which variant types will be recalibrated. This option should be consistent with the one used in the VarCal algorithm.

Alternatively, you can use the option `--vqsr_model` to input a comma-separated list of the required information for multiple VQSR models; this option allows you to apply both a SNP and INDEL mode in a single command line. The syntax of the option is:

```
--vqsr_model var_type=VARIANT_TYPE,\
             recal=VARIANT_RECAL_DATA,\
             tranches_file=TRANCHES_FILE,\
             sensitivity=SENSITIVITY
```

The ApplyVarCal algorithm accepts the following optional ALGO_OPTION:

- `--sensitivity SENSITIVITY`: determine the sensitivity to the available truth sites; only tranches with threshold larger than the sensitivity will be included in the recalibration. We recommend you use a sensitivity number that is included in the tranche threshold list of VarCal algorithm; this will reduce rounding issues. The default value is NULL, so that no tranches filtering is applied, and only the LOW_VQSLOD filter is applied.

8.1.2.23 TNsnv ALGORITHM

The TNsnv algorithm performs the somatic variant calling on the tumor-normal matched pair or the tumor and panel of normal data, using a Genotyper algorithm.

The input to the TNsnv algorithm is a BAM file; its output is a VCF file.

The TNsnv algorithm requires the following ALGO_OPTION:

- `--tumor_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the tumor sample.

Depending on the mode it is run, the TNsnv algorithm may require the following ALGO_OPTION:

- `--normal_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the normal sample.
- `--detect_pon`: indicates that you are using the TNsnv algorithm to create a VCF file that will be part of a panel of normal.
- `--cosmic COSMIC_VCF`: location of the Catalogue of Somatic Mutations in Cancer (COSMIC) VCF file used to create the panel of normal file. Only one file is supported.
- `--pon PANEL_OF_NORMAL_VCF`: location of the file containing the variants detected in the Panel of Normal analysis that will be used to remove false positives. Only one file is supported.

The TNsnv algorithm accepts the following optional ALGO_OPTION:

- `--dbsnp dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP). The variants in the dbSNP will be more likely to be marked as germline as they require more evidence of absence in the normal. Only one file is supported.
- `--call_stats_out CALL_STATS_FILE`: location and filename of the file containing the call stats information from the somatic variant calling.
- `--stdcov_out COVERAGE_FILE`: location and filename of the wiggle file containing the standard coverage.
- `--tumor_depth_out TUMOR_DEPTH_FILE`: location and filename of the wiggle file containing the depth of the tumor sample reads.
- `--normal_depth_out NORMAL_DEPTH_FILE`: location and filename of the wiggle file containing the depth of the normal sample reads.
- `--power_out POWER_FILE`: location and filename of the power file.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 5.
- `--min_init_tumor_lod NUMBER`: minimum tumor log odds in the initial pass calling variants. The default value is 4.
- `--min_tumor_lod NUMBER`: minimum tumor log odds in the final call of variants. The default value is 6.3.
- `--min_normal_lod NUMBER`: minimum normal log odds used to check that the tumor variant is not a normal variant. The default value is 2.2.

- `--contamination_frac NUMBER`: estimation of the contamination fraction from other samples. The default value is 0.02.
- `--min_cell_mutation_frac NUMBER`: minimum fraction of cells which have mutation. The default value is 0.
- `--min_strand_bias_lod NUMBER`: minimum log odds for calling strand bias. The default value is 2.
- `--min_strand_bias_power NUMBER`: minimum power for calling strand bias. The default value is 0.9.
- `--min_dbsnp_normal_lod NUMBER`: minimum log odds for calling normal non-variant at dbsnp sites. The default value is 5.5.
- `--min_normal_allele_frac NUMBER`: minimum allele fraction to be considered in normal; this parameter is useful when the normal sample is contaminated with the tumor sample. The default value is 0.
- `--min_tumor_allele_frac NUMBER`: minimum allelic fraction in tumor sample. The default value is 0.005.
- `--min_normal_allele_frac NUMBER`: minimum allele fraction to be considered in normal; this parameter is useful when the normal sample is contaminated with the tumor sample. The default value is 0.
- `--min_tumor_allele_frac NUMBER`: minimum allelic fraction in tumor sample. The default value is 0.005.
- `--max_indel NUMBER`: maximum nearby indel events that are allowed. The default value is 3.
- `--max_read_clip_frac NUMBER`: maximum fraction of soft/hard clipped bases in a read. The default value is 0.3.
- `--max_mapq0_frac NUMBER`: maximum ratio of reads whose mapq are 0 used to determine poor mapped area. The default value is 0.5.
- `--min_pir_median NUMBER`: minimum read position median. The default value is 10.
- `--min_pir_mad NUMBER`: minimum read position median absolute deviation. The default value is 3.
- `--max_alt_mapq NUMBER`: maximum value of alt allele mapping quality score. The default value is 20.
- `--max_normal_alt_cnt NUMBER`: maximum alt alleles count in normal pileup. The default value is 2.
- `--max_normal_alt_qsum NUMBER`: maximum quality score sum of alt allele in normal pileup. The default value is 20.
- `--max_normal_alt_frac NUMBER`: maximum fraction of alt allele in normal pileup. The default value is 0.03.
- `--power_allele_frac NUMBER`: allele fraction used in power calculations. The default value is 0.3.

8.1.2.24 TNhaplotyper ALGORITHM

The TNhaplotyper algorithm performs the somatic variant calling on the tumor-normal matched pair or the tumor and panel of normal data, using a Haplotyper algorithm.

The input to the TNhaplotyper algorithm is a BAM file; its output is a VCF file.

The TNhaplotyper algorithm requires the following ALGO_OPTION:

- `--tumor_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the tumor sample.

Depending on the mode it is run, the TNhaplotyper algorithm may require the following ALGO_OPTION:

- `--normal_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the normal sample.

- `--detect_pon`: indicates that you are using the TNhaplotyper algorithm to create a VCF file that will be part of a panel of normal.
- `--cosmic COSMIC_VCF`: location of the Catalogue of Somatic Mutations in Cancer (COSMIC) VCF file used to create the panel of normal file. Only one file is supported.
- `--pon PANEL_OF_NORMAL_VCF`: location of the file containing the variants detected in the Panel of Normal analysis that will be used to remove false positives. Only one file is supported.

The TNhaplotyper algorithm accepts the following optional ALGO_OPTION:

- `--dbsnp dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP). The variants in the dbSNP will be more likely to be marked as germline as they require more evidence of absence in the normal. Only one file is supported.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 10.
- `--prune_factor FACTOR`: minimum pruning factor in local assembly; paths with fewer supporting kmers than FACTOR will be pruned from the graph. The default value is 2.
- `--pcr_indel_model MODEL`: PCR indel model used to weed out false positive indels more or less aggressively. The possible modes are‘: NONE (used for PCR free samples), and HOSTILE, AGGRESSIVE and CONSERVATIVE, in order of decreasing aggressiveness. The default value is HOSTILE.
- `--phasing [1/0]`: flag to enable or disable phasing in the output.
- `--min_init_tumor_lod NUMBER`: minimum tumor log odds in the initial pass calling variants. The default value is 4.
- `--min_init_normal_lod NUMBER`: minimum tumor log odds in the initial pass calling variants. The default value is 0.5.
- `--min_tumor_lod NUMBER`: minimum tumor log odds in the final call of variants. The default value is 6.3.
- `--min_normal_lod NUMBER`: minimum normal log odds used to check that the tumor variant is not a normal variant. The default value is 2.2.
- `--min_strand_bias_lod NUMBER`: minimum log odds for calling strand bias. The default value is 2.
- `--min_strand_bias_power NUMBER`: minimum power for calling strand bias. The default value is 0.9.
- `--min_pir_median NUMBER`: minimum read position median. The default value is 10.
- `--min_pir_mad NUMBER`: minimum read position median absolute deviation. The default value is 3.
- `--max_normal_alt_cnt NUMBER`: maximum alt alleles count in normal pileup. The default value is 2.
- `--max_normal_alt_qsum NUMBER`: maximum quality score sum of alt allele in normal pileup. The default value is 20.
- `--max_normal_alt_frac NUMBER`: maximum fraction of alt allele in normal pileup. The default value is 0.03.
- `--tumor_contamination_frac NUMBER`: estimation of the contamination fraction on the tumor sample from other samples. The default value is 0.
- `--normal_contamination_frac NUMBER`: estimation of the contamination fraction on the normal sample from other samples. The default value is 0.
- `--filter_clustered_read_position`: filters variants that are clustered at the start or end of sequencing reads
- `--filter_strand_bias`: filters variants that show evidence of strand bias

- `--bam_output OUTPUT_BAM`: output a BAM file containing modified reads after the local reassembly done by the variant calling. This option should only be used in conjunction with a small bed file for troubleshooting purposes.

8.1.2.25 TNhaplotyper2 ALGORITHM

The TNhaplotyper2 algorithm performs the somatic variant calling on the tumor-normal matched pair or the tumor and panel of normal data, using a Haplotyper algorithm.

The input to the TNhaplotyper2 algorithm is a BAM file; its output is a VCF file.

The TNhaplotyper2 algorithm requires the following ALGO_OPTION:

- `--tumor_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the tumor sample.

Depending on the mode it is run, the TNhaplotyper2 algorithm may require the following ALGO_OPTION:

- `--normal_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the normal sample.
- `--pon PANEL_OF_NORMAL_VCF`: location of the file containing the variants detected in the Panel of Normal analysis that will be used to remove false positives. Only one file is supported.

The TNhaplotyper2 algorithm accepts the following optional ALGO_OPTION:

- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 10.
- `--prune_factor FACTOR`: minimum pruning factor in local assembly; paths with fewer supporting kmers than FACTOR will be pruned from the graph. The default value is 2.
- `--pcr_indel_model MODEL`: PCR indel model used to weed out false positive indels more or less aggressively. The possible modes are‘: NONE (used for PCR free samples), and HOSTILE, AGGRESSIVE and CONSERVATIVE, in order of decreasing aggressiveness. The default value is CONSERVATIVE.
- `--min_init_tumor_lod NUMBER`: minimum tumor log odds in the initial pass calling variants. The default value is 2.0.
- `--min_tumor_lod NUMBER`: minimum tumor log odds in the final call of variants. The default value is 3.0.
- `--min_normal_lod NUMBER`: minimum normal log odds used to check that the tumor variant is not a normal variant. The default value is 2.2.
- `--tumor_contamination_frac NUMBER`: estimation of the contamination fraction on the tumor sample from other samples. The default value is 0.
- `--normal_contamination_frac NUMBER`: estimation of the contamination fraction on the normal sample from other samples. The default value is 0.
- `--germline_vcf VCF`: location of the VCF containing the population allele frequency.
- `--default_af AF`: determines the af value for alleles not found in the germline vcf. The default value is: 0.001.
- `--max_germline_af AF`: determines the maximum germline allele frequency in tumor-only model. The default value is 0.01.
- `--call_pon_sites`: determines whether to call candidate variants even if they are present in the Panel of Normal input.
- `--bam_output OUTPUT_BAM`: output a BAM file containing modified reads after the local reassembly done by the variant calling. This option should only be used in conjunction with a small bed file for troubleshooting purposes.

8.1.2.26 TNscope ALGORITHM

The TNscope algorithm performs the somatic variant calling on the tumor-normal matched pair or the tumor only data, using a Haplotyper algorithm.

The input to the TNscope algorithm is a BAM file; its output is a VCF file.

The TNscope algorithm requires the following ALGO_OPTION:

- `--tumor_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the tumor sample.

The TNscope algorithm accepts the following optional ALGO_OPTION:

- `--normal_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the normal sample. When doing tumor-only somatic calling, this argument is not required.
- `--cosmic COSMIC_VCF`: location of the Catalogue of Somatic Mutations in Cancer (COSMIC) VCF file used to create the panel of normal file. Only one file is supported.
- `--pon PANEL_OF_NORMAL_VCF`: location of the file containing the variants detected in the Panel of Normal analysis that will be used to remove false positives. Only one file is supported. This file is the same as the one used with TNhaplotyper.
- `--dbsnp dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP). The variants in the dbSNP will be more likely to be marked as germline as they require more evidence of absence in the normal. Only one file is supported.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 10.
- `--prune_factor FACTOR`: minimum pruning factor in local assembly; paths with fewer supporting kmers than FACTOR will be pruned from the graph. The default value is 2.
- `--pcr_indel_model MODEL`: PCR indel model used to weed out false positive indels more or less aggressively. The possible modes are': NONE (used for PCR free samples), and HOSTILE, AGGRESSIVE and CONSERVATIVE, in order of decreasing aggressiveness. The default value is HOSTILE.
- `--phasing [1/0]`: flag to enable or disable phasing in the output.
- `--min_init_tumor_lod NUMBER`: minimum tumor log odds in the initial pass calling variants. The default value is 4.
- `--min_init_normal_lod NUMBER`: minimum tumor log odds in the initial pass calling variants. The default value is 0.5.
- `--min_tumor_lod NUMBER`: minimum tumor log odds in the final call of variants. The default value is 6.3.
- `--min_normal_lod NUMBER`: minimum normal log odds used to check that the tumor variant is not a normal variant. The default value is 2.2.
- `--tumor_contamination_frac NUMBER`: estimation of the contamination fraction on the tumor sample from other samples. The default value is 0.
- `--normal_contamination_frac NUMBER`: estimation of the contamination fraction on the normal sample from other samples. The default value is 0.
- `--given GIVEN_VCF`: perform variant calling using only the variants provided in the GIVEN_VCF. The calling will only evaluate the locus and alleles provided in the file.
- `--bam_output OUTPUT_BAM`: output a BAM file containing modified reads after the local reassembly done by the variant calling. This option should only be used in conjunction with a small bed file for troubleshooting purposes.

8.1.2.27 RNASplitReadsAtJunction ALGORITHM

The RNASplitReadsAtJunction algorithm performs the splitting of reads into exon segments by getting rid of Ns but maintaining grouping information, and hard-clipping any sequences overhanging into the intron regions.

The input to the RNASplitReadsAtJunction algorithm is a BAM file; its output is a BAM file.

The RNASplitReadsAtJunction algorithm requires the following ALGO_OPTION:

- `--reassign_mapq IN_QUAL:OUT_QUAL`: the algorithm will reassign mapping qualities from IN_QUAL to OUT_QUAL. This argument is required because STAR assigns a quality of 255 to good alignments instead of the expected default score of 60.

The RNASplitReadsAtJunction algorithm accepts the following optional ALGO_OPTION:

- `--ignore_overhang`: determines whether to ignore and not fix the overhanging sections of the reads.
- `--overhang_max_bases NUMBER`: determines the maximum number of bases allowed in a hard-clipped overhang, so that if there are more bases in the overhang, the overhang will not be hard-clipped. The default value is 40.
- `--overhang_max_mismatches NUMBER`: determines the maximum number of mismatches allowed in a non-hard-clipped overhang, so that the complete overhang will be hard-clipped if the number of mismatches is too high. The default value is 1.
- `--cram_write_options compressor=[gzip|bzip2|lzma|rans]`: CRAM output compression options. compressor=gzip is default if not defined.
- `--cram_write_options version=[2.1|3.0]`: CRAM output version options. version=2.1 is default if not defined.

8.1.2.28 ContaminationAssessment ALGORITHM

The ContaminationAssessment algorithm assesses the contamination present in a sample BAM file; the output of this algorithm can be used as the value of argument contamination_frac, normal_contamination_frac and tumor_contamination_frac in the TNseq and TNscope tools.

The input to the ContaminationAssessment algorithm is a BAM file; its output is a text file.

The ContaminationAssessment algorithm requires the following ALGO_OPTION:

- `--pop_vcf VCF_FILE`: the location of the VCF file containing the allele frequency information for the specific population of the sample.
- `--genotype_vcf VCF_FILE`: the location of the VCF file containing the DNAseq variants reported for the individual; to calculate the contamination in the tumor sample, you should use the DNAseq variants reported for the normal sample. You can create this file by using Haplotyper or Genotyper on the sample bam.

The ContaminationAssessment algorithm accepts the following optional ALGO_OPTION:

- `--type ASSESS_TYPE`: determines the type for the estimate. The possible values are SAMPLE, READGROUP and META to assess the contamination by sample, by lane, or in aggregate across all the reads. Multiple instances of the option are allowed that will assess the contamination at multiple levels. The default value is META.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in the contamination assessment. Any base with quality less than QUALITY will be ignored. The default value is 20.
- `--min_map_qual QUALITY`: determines the filtering quality of the reads used in the contamination assessment. Any read with quality less than QUALITY will be ignored. The default value is 20.

- `--min_basecount NUMBER`: determines the minimum number of bases that need to be present at a locus before the contamination is assessed. The default value is 500.
- `--trim_thresh NUMBER`: threshold that will be used to trim sites; if the probability of the contamination ratio being larger than 0.5 is larger than the threshold, the site will not be included in the contamination assessment. The default value is 0.95.
- `--trim_frac NUMBER`: determines the maximum fraction of sites that may be trimmed based on the trim threshold. The default value is 0.01.
- `--precision NUMBER`: determines the PRECISION on the output percent number. The default value is 0.1.
- `--base_report FILE`: location and filename of the output file that will contain an extended report about the processed data.
- `--population POPULATION_NAME`: a population to use to determine the baseline allele frequency of the sample. The default value is CEU.

8.1.2.29 TNModelApply ALGORITHM

The TNModelApply algorithm applies a Machine Learning model on the results of TNscope to help with variant filtration. This algorithm is only supported in the Linux version of the Sentieon Genomics software.

The TNModelApply algorithm has no input in the driver level, its output is a VCF file.

The TNModelApply algorithm requires the following ALGO_OPTION:

- `-v INPUT`: location of the VCF file from the TNscope variant calling; you can use a VCF file compressed with bgzip and indexed.
- `-m MODEL_FILE`: location of the file containing the Machine Learning model.

The TNModelApply algorithm modifies the input VCF file by adding the MLrejected FILTER to the variants; since the FILTER is added, you may want to remove any FILTERs already present in the input VCF, as they may no longer be relevant. You can use bcftools(https://samtools.github.io/bcftools/bcftools.html) for that purpose:

```
$BCF/bcftools annotate -x "^FILTER/MLrejected,FILTER/PASS" -O z \
  -o $OUTPUT.vcf.gz $INPUT.vcf.gz
```

8.1.2.30 DNAscope ALGORITHM

The DNAscope algorithm performs an improved version of Haplotype variant calling.

The input to the DNAscope algorithm is a BAM file; its output is a VCF file.

The DNAscope algorithm accepts the following optional ALGO_OPTION:

- `--annotation 'ANNOTATION_LIST'`: determines additional annotations that will be added to the output VCF. Use a comma separated list to enable or disable annotations. Include 'none' to remove the default annotations; prefix annotations with the exclamation point (!) to disable the specific annotation. See Section 9.4 for supported annotations.
- `-d dbSNP_FILE`: location of the Single Nucleotide Polymorphism database (dbSNP) used to label known variants. Only one file is supported.
- `--var_type VARIANT_TYPE`: determine which variant types will be called; VARIANT_TYPE is a comma separated list of the following possible values:

  - SNP to call Single Nucleotide Polymorphism. This is included in the default behavior.
  - INDEL to call insertion-deletions. This is included in the default behavior.
  - BND to call break-end information required for the structural variant caller.
- `--call_conf CONFIDENCE`: determine the threshold of variant quality to call a variant. Variants with quality less than CONFIDENCE will be removed.
- `--emit_conf CONFIDENCE`: determine the threshold of variant quality to emit a variant. Variants with quality less than CONFIDENCE will be not be added to the output VCF file.
- `--emit_mode MODE`: determines what calls will be emitted. Possible values for mode are:
  - variant: emit calls only at confident variant sites. This is the default behavior.
  - confident: emit calls at confident variant sites or confident reference sites.
  - all: emit all calls, regardless of their confidence.
  - gvcf: emits additional information required for joint calling. This option is required if you want to perform joint calling using the GVCFtyper algorithm.
- `--gq_bands LIST_OF_BANDS`: determines the bands that will be used to compress variants of similar genotype quality (GQ) that will be emitted as a single VCF record in the GVCF output file. The LIST_OF_BANDS is a comma-separated list of bands where each band is defined by START-END/STEP. The default is 1-60,60-99/10,99.
- `--min_base_qual QUALITY`: determines the filtering quality of the bases used in variant calling. Any base with quality less than QUALITY will be ignored. The default value is 10.
- `--pcr_indel_model MODEL`: PCR indel model used to weed out false positive indels more or less aggressively. The possible MODELs are: NONE (used for PCR free samples), and HOSTILE, AGGRESSIVE and CONSERVATIVE, in order of decreasing aggressiveness. The default value is CONSERVATIVE.
- `--phasing [1/0]`: flag to enable or disable phasing in the output when using emit*mode GVCF. The default value is 1 (on) and this flag has no impact when using an emit*mode other than GVCF. Phasing is only calculated for diploid samples.
- `--ploidy PLOIDY`: determines the ploidy number of the sample being processed. The default is 2.
- `--prune_factor FACTOR`: minimum pruning factor in local assembly; paths with fewer supporting kmers than FACTOR will be pruned from the graph. The default value is 2.
- `--trim_soft_clip`: determines whether soft clipped bases in the reads should be included in the variant calling. This argument is only recommended to process RNA reads.
- `--given GIVEN_VCF`: perform variant calling using only the variants provided in the GIVEN_VCF. The calling will only evaluate the locus and alleles provided in the file. This option cannot be used in conjunction with `--emit_mode gvcf`.
- `--bam_output OUTPUT_BAM`: output a BAM file containing modified reads after the local reassembly done by the variant calling. This option should only be used in conjunction with a small bed file for troubleshooting purposes.
- `-- filter_chimeric_reads`: determines whether chimeric reads will be used when calling variants. The default is to include chimeric reads only if the `var_type BND` is set.
- `--snp_heterozygosity HETEROZIGOSITY`: determines the expected heterozygosity value used to compute prior likelihoods in SNP calling.
- `--indel_heterozygosity HETEROZIGOSITY`: determines the expected heterozygosity value used to compute prior likelihoods in INDEL calling.

- `--model MODEL_FILE`: the location of the machine learning model file that will be used with the DNAModelApply tool; the model will be used to determine the settings used in variant calling.

8.1.2.31 DNAModelApply ALGORITHM

The DNAModelApply algorithm performs the second step of variant calling using DNAscope. This algorithm is only supported in the Linux version of the Sentieon Genomics software.

The DNAModelApply algorithm has no input in the driver level, its output is a VCF containing the variants.

The DNAModelApply algorithm requires the following ALGO_OPTION:

- `-v INPUT`: location of the VCF file from the DNAscope variant calling algorithm performed with an input model file determining the correct settings. You can use VCF files compressed with bgzip and indexed.
- `-m MODEL_FILE`: the location of the machine learning model file; this file should be the same as the one used in the DNAscope command to generate the input VCF.

The DNAModelApply algorithm does not support any optional ALGO_OPTION.

The DNAModelApply algorithm modifies the input VCF file by adding the MLrejected FILTER to the variants; since the FILTER is added, you may want to remove any FILTERs already present in the input VCF, as they may no longer be relevant. You can use bcftools(https://samtools.github.io/bcftools/bcftools.html) for that purpose:

```
$BCF/bcftools annotate -x "^FILTER/MLrejected,FILTER/PASS" -O z \
  -o $OUTPUT.vcf.gz $INPUT.vcf.gz
```

8.1.2.32 SVSolver ALGORITHM

The SVSolver algorithm performs the structural variant calling of a sample, provided that the sample has been previously processed using the DNAscope algorithm with the option `--var_type bnd`.

The SVSolver algorithm has no input in the driver level, its output is a VCF containing the called structural variants.

The SVSolver algorithm requires the following ALGO_OPTION:

- `-v INPUT`: location of the VCF file from the DNAscope variant calling algorithm performed on a sample using the option `--var_type bnd`. You can use VCF files compressed with bgzip and indexed.

The SVSolver algorithm does not support any optional ALGO_OPTION.

8.1.3 DRIVER read_filter options

The `--read_filter` argument of the DRIVER binary allows to filter or transform reads from the input BAM file before performing the calculation.

The syntax for the argument is: `--read_filter FILTER,OPTION=VALUE,OPTION=VALUE,. . .`

8.1.3.1 QualCalFilter read_filter

The QualCalFilter read filter is used to transform reads and perform base quality score recalibration while modifying the information contained in the recalibration table.

The QualCalFilter read filter requires one of the following OPTION:

- table=TABLE_FILEPATH: the location of the recalibration table that will be used as the basis to perform the base quality score recalibration.
- use_oq=[true/false]: determines whether to use the original base quality scores contained in the OQ tag in the BAM file. This option cannot be used in conjunction with the table option.

The QualCalFilter read filter accepts the following optional OPTION:

- prior=PRIOR: determines the global bias for all the base quality scores.
- min_qual=QUAL: determines the quality threshold to perform recalibration; bases with quality scores less than QUAL will not be recalibrated.
- levels=LEVEL1/LEVEL2/. . . : determines the static quantization levels of the base quality scores.
- indel=[true/false]: determines whether to add the base quality scores for INDELs into the BAM tags.
- keep_oq=[true/false]: determines whether to keep the original before recalibration base quality scores by using the OQ tag in the bam file.

8.1.3.2 MapQualFilter read_filter

The MapQualFilter read filter is used to filter reads by mapping quality.

The MapQualFilter read filter accepts the following optional OPTION:

- `min_map_qual=QUALITY`: filter out reads with mapping quality less than QUALITY.

8.1.3.3 OverclippingFilter read_filter

The OverclippingFilter read filter is used to filter reads depending on their soft clipping characteristics.

The OverclippingFilter read filter accepts the following optional OPTION:

- `min_align_len=LENGTH`: filter reads where the number of bases that are not soft clipped is less than LENGTH.
- `count_both_ends=[true/false]`: if set to true, only filter reads where both ends of the read are soft clipped, so that reads with soft-clipping on one end only will not be filtered regardless of their non soft-clipped length.

8.2 UTIL binary

UTIL is the binary used to run some utility functions. This binary is mainly used to process the raw reads output from BWA.

8.2.1 UTIL syntax

The general syntax of the UTIL binary is:

```
sentieon util MODE [OPTIONS]
```

The supported modes (**MODE**) for this command are:

- index: build the index for a BAM file. The following command will generate a bai BAM index file at the same location as the input file:

```
sentieon util index INPUT.bam
```

- vcfindex: build the index for a VCF file. The following command will generate a idx VCF index file at the same location as the input file:

```
sentieon util vcfindex INPUT.vcf
```

- sort: sort a BAM file. The optional arguments (OPTIONS) for the UTIL command using the sort ALGORITHM include:
  - `-t NUMBER_THREADS`: number of computing threads that will be used by the software to run parallel processes.
  - **`-r REFERENCE`: location of the reference FASTA file. This argument** is required if you are using a CRAM output file, otherwise it is optional.
  - `-i INPUT`: location of the input file.
  - `-o OUTPUT`: the location and filename of the output file.
  - `--temp_dir DIRECTORY`: determines where the temporary files will be stored. The default is the folder where the command is run (`$PWD`).
  - `--cram_read_options decode_md=0`: CRAM input option to turn off the NM/MD tag in the input CRAM.
  - `--cram_write_options compressor=[gzip|bzip2|lzma|rans]`: CRAM output compression options. compressor=gzip is default if not defined.
  - `--cram_write_options version=[2.1|3.0]`: CRAM output version options. version=2.1 is default if not defined.
  - `--bam_compression COMPRESSION_LEVEL[0-9]`: gzip compression level for the output BAM file. The default value is 6.
  - `--skip_no_coor`: determines whether to skip unmapped reads.
  - `--sam2bam`: indicates that the input will be in the form of an uncompressed SAM file, that needs to be converted to BAM. If this option is not used, the input should have been converted to BAM format from the BWA output using samtools.
  - `--block_size BLOCK_SIZE`: size of the block to be used for sorting.

```
sentieon util sort -t NUMBER_THREADS --sam2bam -i INPUT.sam -o OUTPUT.bam
```

- vcfconvert: compress and decompress VCF and GVCF files.

  The following command will compress and index the input file:

```
sentieon util vcfconvert INPUT.vcf OUTPUT.vcf.gz
```

  The following command will decompress a non-indexed vcf file generated with gzip and then compress and index the file. When using this command make sure that the INPUT and OUTPUT files are not the same:

```
gzip -dc INPUT.vcf.gz | sentieon util vcfconvert - OUTPUT.vcf.gz
```

- stream: perform base quality correction in streaming mode.

  The following command will apply the recalibration and output the corresponding recalibrated BAM file to stdout:

```
sentieon util stream -i INPUT.bam -q RECAL_TABLE \
  --output_index_file OUTPUT.bam.bai -o -
```

The above command will not generate an index file unless the option `output_index_file` is included.

8.3 TNHAPFILTER script

TNHAPFILTER is a script used to filter the results of the TNhaplotyper2 algorithm.

8.3.1 TNHAPFILTER syntax

The general syntax of the TNHAPFILTER script is:

```
sentieon tnhapfilter --tumor_sample TUMOR_SAMPLE_NAME -v TMP_OUT_TN_VCF
  OUT_TN_VCF
```

The following inputs are required for the command:

- TUMOR_SAMPLE_NAME: sample name used for tumor sample in Map reads to reference stage.
- TMP_OUT_TN_VCF: the location and file name of the file containing the unfiltered variants produced by TNhaplotyper2.
- OUT_TN_VCF: the location and file name of the output file containing the variants.

The arguments (OPTIONS) for this command include:

- `-v INPUT__VCF`: location of the VCF file from the DNAscope variant calling algorithm performed with an input model file determining the correct settings. You can use VCF files compressed with bgzip and indexed.
- `--tumor_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the tumor sample.
- `--normal_sample SAMPLE_NAME`: name of the SM tag in the BAM file for the normal sample, if present.
- `--min_tumor_lod NUMBER`: minimum tumorLOD. The default is 5.3.
- `--max_germline_prob NUMBER`: maximum germline probability. The default is 0.025.
- `--max_normal_art_lod NUMBER`: maximum normal artifact LOD. The default is 0.0.
- `--max_alt_cnt NUMBER`: maximum ALT allele count. The default is 1.
- `--max_event_cnt NUMBER`: maximum events in region. The default is 2.
- `--min_median_base_qual NUMBER`: minimum median base quality. The default is 20.
- `--min_median_mapq NUMBER`: minimum median mapping quality. The default is 30.
- `--max_diff_fraglen NUMBER`: maximum median difference in fragment length. THE default is 10000.
- `--min_pir_median NUMBER`: minimum median read position. The default is 5.
- `--max_strand_prob NUMBER`: maximum strand artifact probability. The default is 0.99.
- `--min_strand_af NUMBER`: minimum strand artifact allele fraction. The default is 0.01.
- `--contamination NUMBER`: contamination fraction to filter; if NUMBER is greater than 0, the tool will try to remove a fraction of the reads supporting each ALT. The default is 0.0.

If you are using Python 2.6.x, you may get the following error when running tnhapfilter: `ImportError: No module named argparse`. If that is the case, you will need to install the argparse module to your python installation; you can do this by running `pip install argparse` or whichever other package manager you use.

8.4 PLOT script

PLOT is a script used to create plots of the results of the metrics and recalibration stages. The plots are stored in a PDF file.

8.4.1 PLOT syntax

The general syntax of the PLOT script is:

```
sentieon plot STAGE -o OUTPUPT_FILE INPUTS [OPTIONS]
```

The supported modes (STAGE) for this command are:

- GCBias: generate PDF file from the metrics results of GCBias.
- QualDistribution: generate PDF file from the metrics results of QualDistribution.
- InsertSizeMetricAlgo: generate PDF file from the metrics results of InsertSizeMetricAlgo.
- MeanQualityByCycle: generate PDF file from the metrics results of MeanQualityByCycle.
- QualCal: generate PDF file from the BQSR QualCal tool.
- VarCal: generate PDF file from the VQSR VarCal tool.

8.4.1.1 PLOT results of GCBias STAGE

The INPUTS to generate the plots from the GCBias stage are:

- GC_METRIC_TXT: where GC_METRIC_TXT is the output file of the GCBias algorithm from Section 8.1.2.7.

The plotting of the GCBias metrics STAGE does not accept any OPTIONS.

8.4.1.2 PLOT results of MeanQualityByCycle STAGE

The INPUTS to generate the plots from the MeanQualityByCycle stage are:

- MQ_METRIC_TXT: where MQ_METRIC_TXT is the output file of the MeanQualityByCycle algorithm from Section 8.1.2.5.

The plotting of the MeanQualityByCycle metrics STAGE does not accept any OPTIONS.

8.4.1.3 PLOT results of QualDistribution STAGE

The INPUTS to generate the plots from the QualDistribution stage are:

- QD_METRIC_TXT: where QD_METRIC_TXT is the output file of the QualDistribution algorithm from Section 8.1.2.6.

The plotting of the QualDistribution metrics STAGE does not accept any OPTIONS.

8.4.1.4 PLOT results of InsertSizeMetricAlgo STAGE

The INPUTS to generate the plots from the InsertSizeMetricAlgo stage are:

- IS_METRIC_TXT: where IS_METRIC_TXT is the output file of the InsertSizeMetricAlgo algorithm from Section 8.1.2.10.

The plotting of the InsertSizeMetricAlgo metrics STAGE does not accept any OPTIONS.

8.4.1.5 PLOT results of QualCal STAGE

The INPUTS to generate the plots from the bqsr stage are:

- RECAL_RESULT.CSV: the output csv file of the QualCal algorithm from Section 8.1.2.4.

The plotting of the bqsr STAGE does not accept any OPTIONS.

8.4.1.6 PLOT results of VarCal STAGE

The INPUTS to generate the plots from the vqsr stage is:

- PLOT_FILE: a file created by the VarCal algorithm from Section 8.1.2.21 containing the data required to create the report.

The plotting of the vqsr STAGE accept the following OPTIONS:

- tranches_file=TRANCHES_FILE: location of the file containing the partition of the call sets into quality tranches, generated by the VarCal algorithm from Section 8.1.2.21.
- target_titv=TITV_THRES: expected TiTv number for the species; it is used calculate the True Positive and False Positive numbers in the plot.
- min_fp_rate=MIN_RATE: minimum False Positive number; it is used calculate the True Positive and False Positive numbers in the plot.

8.5 LICSRVR binary

LICSRVR is the binary used to run the license server to facilitate dynamic license assignment and record license utilization within a cluster.

8.5.1 LICSRVR syntax

The general syntax of the LICSRVR binary is:

```
<SENTIEON_FOLDER>/bin/sentieon licsrvr [--start|--stop] [--log LOG_FILE] LICENSE_FILE
```

The following inputs are optional for the command:

- **LOG_FILE**: location and filename of the output file containing the log of the server.
- **LICENSE_FILE**: location of the server license file.

After the license server is operational, the client applications can request license tokens from the server by setting the SENTIEON_LICENSE environment variable to the server address in the form of HOST:PORT.

The licsrvr binary supports the following additional modes:

- `--version`: will report the software package version the binary belongs to.
- `--dump`: will report the current status of the license server, including the number of available licenses.

  ```
  <SENTIEON_FOLDER>/bin/sentieon licsrvr --dump LICENSE_FILE
  ```

- `--dump=update`: if the license information has been updated and automatically pulled by the license server, this mode will dump the updated license information to the stdout. The following command will report the updated license information the license server has, if there has been any change:

  ```
  <SENTIEON_FOLDER>/bin/sentieon licsrvr --dump=update LICENSE_FILE
  ```

8.6 LICCLNT binary

LICCLNT is the binary used to test the license server functionality to help determine whether the license server is operational, and how many licenses of the different algorithms are available.

8.6.1 LICCLNT syntax

The LICCLNT binary has two modes, one to ping the license server and one to check the available licenses for specific algorithms.

You can run the following command to check if the license server is operational:

```
<SENTIEON_FOLDER>/bin/sentieon licclnt ping --server HOST:PORT
```

The command will return 0 if the server is operational.

You can run the following command to check how many licenses are:

```
<SENTIEON_FOLDER>/bin/sentieon licclnt query --server HOST:PORT FEATURE
```

The command will return the number of licenses that are available for the specific license feature, which can be used for managing your jobs: before submitting a job on a certain number of threads, you can check if there are enough licenses for those threads, preventing the tool from being idle while waiting for licenses.

8.7 BWA binary

The BWA binary has two modes of interest, “mem” mode to align FASTQ files against a reference FASTA file, and “shm” mode to load the FASTA index file in memory to be shared among multiple BWA processes running in the same server.

8.7.1 BWA mem syntax

You can run the following command to align a single-ended FASTQ1 file or a pair-ended set of 2 FASTQ files against the FASTA reference, which will produce the mapped reads to stdout, to be piped onto util sort:

```
<SENTIEON_FOLDER>/bin/sentieon bwa mem OPTIONS FASTA FASTQ1 [FASTQ2]
```

The arguments (OPTIONS) for this command include:

- `-t NUMBER_THREADS`: number of computing threads that will be used by the software to run parallel processes. The argument is optional; if omitted the bwa binary will use 1 thread.
- `-p`: determines whether the first input FASTQ file contains interleaved pair-ended reads. If this argument is used, only use a single FASTQ input, as the second FASTQ2 file will be ignored.
- `-M`: determines whether to make split reads as secondary.
- `-R READGROUP_STRING`: Read Group header line that all reads will be attached to. The recommended READGROUP_STRING is `@RG\tID:$readgroup\tSM:$sample\tPL:$platform\tPU:$platform_unit`
  - $readgroup is a unique ID that identifies the reads.
  - $sample is the name of the sample the reads belong to.
  - $platform is the sequencing technology, typically ILLUMINA.
  - $platform_unit is the sequencing element that performed the sequencing.
- `-K CHUNK_SIZE`: determines the size of the group of reads that will be mapped at the same time. If this argument is not set, the results will depend on the number of threads used.

8.7.2 BWA shm syntax

You can run the following command to load the FASTA index file in memory:

```
<SENTIEON_FOLDER>/bin/sentieon bwa shm FASTA
```

You can run the following command to list FASTA indices files stored in memory:

```
<SENTIEON_FOLDER>/bin/sentieon bwa shm -l
```

You can run the following command to remove all FASTA indices files stored in memory, thus freeing memory when no longer necessary:

```
<SENTIEON_FOLDER>/bin/sentieon bwa shm -d
```

The arguments (OPTIONS) for this command include:

- `-t NUMBER_THREADS`: number of computing threads that will be used by the software to run parallel processes. The argument is optional; if omitted the bwa binary will use 1 thread.
- `-f FILE`: location of a temporary file that will be used to reduce peak memory usage.

8.7.3 Controlling memory usage in BWA

By default BWA will use about 24 GB in a Linux system and 8 GB in a Mac system. You can control the memory usage via the bwt_max_mem environment variable, which can be used to limit the memory usage at the expense of speed performance, or enhance the speed performance by using more memory. For example, you will get faster alignment by adding the following to your scripts:

```
export bwt_max_mem=50G
```

Bear in mind that the number you use in the `bwt_max_mem` environmental variable is not a hard limit, but an estimate of the memory used in BWA; as such, it will not guarantee that BWA will use less memory than the provided value.

8.7.4 Using an existing BAM file as input

If you do not have access to the FASTQ inputs, but only have an already aligned and sorted BAM file, you can use it as input and redo the alignment by running *samtools*:

```
samtools collate -@ 32 -Ou INPUT_BAM tmp- | samtools fastq -@ 32 -s \
/dev/null -0 /dev/null - | <SENTIEON_FOLDER>/bin/sentieon bwa mem -t 32 -R \
'@RG\tID:id\tLB:lib\tSM:sample\tPL:ILLUMINA' -M -K 1000000 -p $ref /dev/stdin \
| <SENTIEON_FOLDER>/bin/sentieon util sort -t 32 -o OUTPUT_BAM --sam2bam -
```

Alternatively, you could first create the FASTQ files and then process them as you would normally do:

```
samtools collate -n -@ 32 -uO INPUT_BAM tmp- | samtools fastq -@ 32 \
-s >(gzip -c > single.fastq.gz) -0 >(gzip -c > unpaired.fastq.gz) \
-1 >(gzip -c > output_1.fastq.gz) -2 >(gzip -c > output_2.fastq.gz) -
```

If you do this, you may encounter an abnormal memory usage in BWA; if that is the case, you can follow the instructions in *BWA uses an abnormal amount of memory when using FASTQ files created from a BAM file* .

9

Examples of tool capabilities and applications

9.1 DNA pipeline example script

Below is an example of how to process a set of 2 fastq files for a sample and perform variant calling according to the Broad institute best practices described in https://www.broadinstitute.org/gatk/guide/best-practices.

```
#!/bin/bash
# ******************************************
# 0. input locations
# ******************************************
# Update with the fullpath location of your sample fastq
fastq_folder=”/home/pipeline/samples”
fastq_1=”1.fastq.gz”
fastq_2=”2.fastq.gz” #If using Illumina paired data
sample=”sample_name”
group=”read_group_name”
platform=”ILLUMINA”

# Update with the location of the reference data files
regions=”/home/ref_hg19/TruSeq_exome_regions.hg19.bed”
fasta=”/home/ref_hg19/ucsc.hg19.fasta”
dbsnp=”/home/ref_hg19/dbsnp_138.hg19.vcf.gz”
known_sites=”/home/ref_hg19/Mills_and_1000G_gold_standard.indels.hg19.sites.vcf.gz”

# Update with the location of the Sentieon software package and license file
release_dir=/home/release/sentieon-genomics-201711.03
export SENTIEON_LICENSE=/home/Licenses/regression.lic

# Other settings
nt=16 #number of threads to use in computation
workdir=”$HOME/testcase” #Determine where the output files will be stored
run_joint=”no”

# ******************************************
# 0. Setup
# ******************************************
mkdir -p $workdir
logfile=$workdir/run.log
exec >$logfile 2>&1
```

(continues on next page)

(continued from previous page)

```
cd $workdir

# ******************************************
# 1. Mapping reads with BWA-MEM, sorting
# ******************************************
#The results of this call are dependent on the number of threads used.
#To have number of threads independent results, add chunk size option -K 10000000
($release_dir/bin/sentieon bwa mem -M -R "@RG\tID:$group\tSM:$sample\tPL:$pl” \
  -t $nt $fasta $fastq_1 $fastq_2 || echo -n 'bwa error') | \
  $release_dir/bin/sentieon util sort -o sorted.bam -t $nt --sam2bam -i -

# ******************************************
# 2. Metrics
# ******************************************
$release_dir/bin/sentieon driver -r $fasta -t $nt -i sorted.bam \
  --algo MeanQualityByCycle mq_metrics.txt \
  --algo QualDistribution qd_metrics.txt \
  --algo GCBias --summary gc_summary.txt gc_metrics.txt \
  --algo AlignmentStat aln_metrics.txt \
  --algo InsertSizeMetricAlgo is_metrics.txt
$release_dir/bin/sentieon plot GCBias -o gc-report.pdf gc_metrics.txt
$release_dir/bin/sentieon plot QualDistribution -o qd-report.pdf qd_metrics.txt
$release_dir/bin/sentieon plot MeanQualityByCycle -o mq-report.pdf mq_metrics.txt
$release_dir/bin/sentieon plot InsertSizeMetricAlgo -o is-report.pdf is_metrics.txt

# ******************************************
# 3. Remove Duplicate Reads
# ******************************************
$release_dir/bin/sentieon driver -t $nt -i sorted.bam \
  --algo LocusCollector --fun score_info score.txt
$release_dir/bin/sentieon driver -t $nt -i sorted.bam \
  --algo Dedup --rmdup --score_info score.txt --metrics dedup_metrics.txt deduped.bam

# ******************************************
# 4. Indel realigner
# ******************************************
$release_dir/bin/sentieon driver -r $fasta -t $nt -i deduped.bam \
  --algo Realigner -k $known_sites --interval_list $regions realigned.bam

# ******************************************
# 5. Base recalibration
# ******************************************
$release_dir/bin/sentieon driver -r $fasta -t $nt -i realigned.bam \
  --interval $regions --algo QualCal -k $dbsnp -k $known_sites recal_data.table
$release_dir/bin/sentieon driver -r $fasta -t $nt -i realigned.bam \
  --interval $regions -q recal_data.table \
  --algo QualCal -k $dbsnp -k $known_sites recal_data.table.post
$release_dir/bin/sentieon driver -t $nt --algo QualCal \
  --plot --before recal_data.table \
  --after recal_data.table.post recal.csv
$release_dir/bin/sentieon plot QualCal -o recal_plots.pdf recal.csv

# ******************************************
# 6b. HC Variant caller
# ******************************************
if [ "$run_joint” == "yes” ]; then
```

(continues on next page)

(continued from previous page)

```
 gvcf_option="--emit_mode gvcf"
fi
$release_dir/bin/sentieon driver -r $fasta --interval $regions -t $nt \
  -i realigned.bam -q recal_data.table --algo Haplotyper -d $dbsnp \
  --emit_conf=30 --call_conf=30 $gvcf_option output-hc.vcf.gz

# ******************************************
# 5b. ReadWriter to output recalibrated bam
# This stage is optional as variant callers
# can perform the recalibration on the fly
# using the before recalibration BAM plus
# the recalibration table
# This stage should not include interval
# option to prevent read loss
# ******************************************
$release_dir/bin/sentieon driver -r $fasta -t $nt -i realigned.bam \
  -q recal_data.table --algo ReadWriter recaled.bam
```

9.2 Example script to perform VQSR

Below is an example of how to calculate, apply and create a report for the Variant Quality Score Recalibration (VQSR).

```
# ******************************************
# 7. Variant Recalibration
# ******************************************
# Update with the location of the resource files for VQSR
dbsnp="/home/pipeline/ref_hg19/dbsnp_138.hg19.vcf.gz"
dbsnp_Mill="/home/ref_hg19/Mills_and_1000G_gold_standard.indels.hg19.sites.vcf.gz"
dbsnp_1000G_omni="/home/ref_hg19/1000G_omni2.5.hg19.sites.vcf.gz"
dbsnp_hapmap="/home/ref_hg19/hapmap_3.3.hg19.sites.vcf.gz"
dbsnp_1000G_phase1="/home/ref_hg19/1000G_phase1.snps.high_confidence.hg19.sites.vcf.gz"
dbsnp_1000G_phase1_indel="/home/ref_hg19/1000G_phase1.indel.hg19.sites.vcf.gz"

#first for SNPs
#create the resource argument
resource_text="--resource $dbsnp_1000G_phase1 \
              --resource_param 1000G,known=false,training=true,truth=false,prior=10.0 "
resource_text="$resource_text --resource $dbsnp_1000G_omni \
              --resource_param omni,known=false,training=true,truth=true,prior=12.0 "
resource_text="$resource_text --resource $dbsnp \
              --resource_param dbsnp,known=true,training=false,truth=false,prior=2.0 "
resource_text="$resource_text --resource $dbsnp_hapmap \
              --resource_param hapmap,known=false,training=true,truth=true,prior=15.0"
#create the annotation argument
annotation_array="DP QD FS SOR MQ MQRankSum ReadPosRankSum InbreedingCoeff"
for annotation in $annotation_array; do
 annotate_text="$annotate_text --annotation $annotation"
done
#Run the VQSR
$release_dir/bin/sentieon driver -r $REFERENCE --algo VarCal -v output-hc.vcf.gz \
  $resource_text $annotate_text --var_type SNP --plot_file vqsrSNP.hc.plotfile \
  --tranches_file vqsrSNP.hc.tranches vqsrSNP.hc.recal
#apply the VQSR
```

(continues on next page)

(continued from previous page)

```
$release_dir/bin/sentieon driver -r $REFERENCE --algo ApplyVarCal -v output-hc.vcf.gz \
  --var_type SNP --tranches_file vqsrSNP.hc.tranches --sensitivity 99.0 \
  --recal vqsrSNP.hc.recal vqsrSNP.hc.recaled.vcf.gz
#plot the report
$release_dir/bin/sentieon plot vqsr -o vqsrSNP.pdf vqsrSNP.hc.plotfile

#for indels after SNPs
#create the resource argument
resource_text="--resource $dbsnp_1000G_phase1_indel \
              --resource_param 1000G,known=false,training=true,truth=false,prior=10.0 "
resource_text="$resource_text --resource $dbsnp_Mill \
              --resource_param Mills,known=false,training=true,truth=true,prior=12.0 "
resource_text="$resource_text --resource $dbsnp \
              --resource_param dbsnp,known=true,training=false,truth=false,prior=2.0 "
#create the annotation argument
annotation_array="QD DP FS SOR MQ ReadPosRankSum"
annotate_text=""
for annotation in $annotation_array; do
  annotate_text="$annotate_text --annotation $annotation"
done
#Run the VQSR
$release_dir/bin/sentieon driver -r $REFERENCE --algo VarCal -v output-hc.vcf.gz \
  $resource_text $annotate_text --var_type INDEL --plot_file vqsrINDEL.hc.plotfile \
  --max_gaussians 4 --tranches_file vqsrINDEL.hc.tranches vqsrINDEL.hc.recal
#apply the VQSR
$release_dir/bin/sentieon driver -r $REFERENCE --algo ApplyVarCal -v vqsrSNP.hc.recaled.vcf.gz \
  --var_type INDEL --recal vqsrINDEL.hc.recal --tranches_file vqsrINDEL.hc.tranches \
  --sensitivity 99.0 vqsrSNPINDEL.hc.recaled.vcf.gz
#plot the report
$release_dir/bin/sentieon plot vqsr -o vqsrINDEL.VQSR.pdf vqsrINDEL.hc.plotfile
```

9.3 Working with multiple input files

Typically, there will be two occasions when you will be having multiple input files that need to be processed together: a single sample has been sequenced in multiple lanes, or you are processing multiple samples from a cohort.

9.3.1 Working with multiple input files from the same sample

When you have more than one set of input fastq files for the sample, you should perform an individual alignment stage for each set of input fastq files, and then use the multiple sorted BAM files as input of the next stage. The next stage will take care of merging all the BAM files into one. You need to make sure that each set of input fastq files are aligned using the same SM sample name, but a different RG readgroup name, so that the rest of the stages properly deal with the different data.

```
#Run alignment for 1st input file set
(sentieon bwa mem -M -R '@RG\tID:GROUP_NAME_1 \tSM:SAMPLE_NAME \tPL:PLATFORM' \
  -t NUMBER_THREADS REFERENCE SAMPLE_1 [SAMPLE_1_2] || echo -n 'error' ) \
  | sentieon util sort -o SORTED_BAM_1 -t NUMBER_THREADS --sam2bam -i -
#Run alignment for 2nd input file set
(sentieon bwa mem -M -R '@RG\tID:GROUP_NAME_2 \tSM:SAMPLE_NAME \tPL:PLATFORM' \
  -t NUMBER_THREADS REFERENCE SAMPLE_2 [SAMPLE_2_2] || echo -n 'error' \
```

(continues on next page)

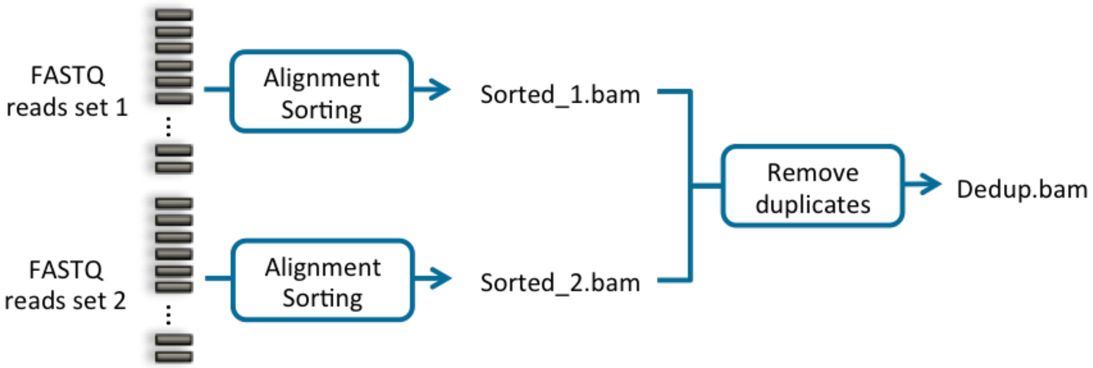


Fig. 9.1: Bioinformatics pipeline using multiple input files from the same sample

(continued from previous page)

```
  sentieon util sort -o SORTED_BAM_2 -t NUMBER_THREADS --sam2bam -i -
#Run dedup on both BAM files
sentieon driver -t NUMBER_THREADS -i SORTED_BAM_1 -i SORTED_BAM_2 \
  --algo LocusCollector --fun score_info SCORE_TXT
sentieon driver -t NUMBER_THREADS -i SORTED_BAM_1 -i SORTED_BAM_2 \
  --algo Dedup --rmdup --score_info SCORE_TXT --metrics DEDUP_METRIC_TXT DEDUP_BAM
```

Alternatively you can input the results of multiple BWA alignments onto a single util sort command to merge the results onto a single sorted bam files as soon as possible. You still need to make sure that each set of input fastq files are aligned using the same SM sample name, but a different RG readgroup name, so that the rest of the stages properly deal with the different data.

```
#Run alignment for both input file sets sequentially
(sentieon bwa mem -M -R '@RG\tID:GROUP_NAME_1\tSM:SAMPLE_NAME\tPL:PLATFORM' \
  -t NUMBER_THREADS REFERENCE SAMPLE_1 [SAMPLE_1_2]  && sentieon bwa mem -M -R \
  '@RG\tID:GROUP_NAME_2 \tSM:SAMPLE_NAME \tPL:PLATFORM' \
  -t NUMBER_THREADS REFERENCE SAMPLE_2 [SAMPLE_2_2] || echo -n 'error' ) \
  | sentieon util sort -o SORTED_COMBINED_BAM -t NUMBER_THREADS --sam2bam -i -
#Run dedup on both BAM files
sentieon driver -t NUMBER_THREADS -i SORTED_COMBINED_BAM_1 \
  --algo LocusCollector --fun score_info SCORE_TXT
sentieon driver -t NUMBER_THREADS -i SORTED_COMBINED_BAM_1 \
  --algo Dedup --rmdup --score_info SCORE_TXT --metrics DEDUP_METRIC_TXT DEDUP_BAM
```

9.3.2 Working with multiple input files from different samples

Processing multiple samples from a cohort together, also known as joint variant calling, can be done in two ways:

- Process each sample individually and create either a recalibrated BAM file or a realigned BAM and a recalibration table for each sample, then process all files using the Haplotyper algorithm.

```
#Process all samples independently until BQSR
for sample in s1 s2 s3; do
  (sentieon bwa mem -M -R '@RG\tID:GROUP_${sample}\tSM:${sample}\tPL:PLATFORM' \
    -t NUMBER_THREADS REFERENCE ${sample}_FASTQ_1 ${sample}_FASTQ_2 || echo -n \
    'error' )| sentieon util sort -o ${sample}_SORTED_BAM -t NUMBER_THREADS \
```

(continues on next page)

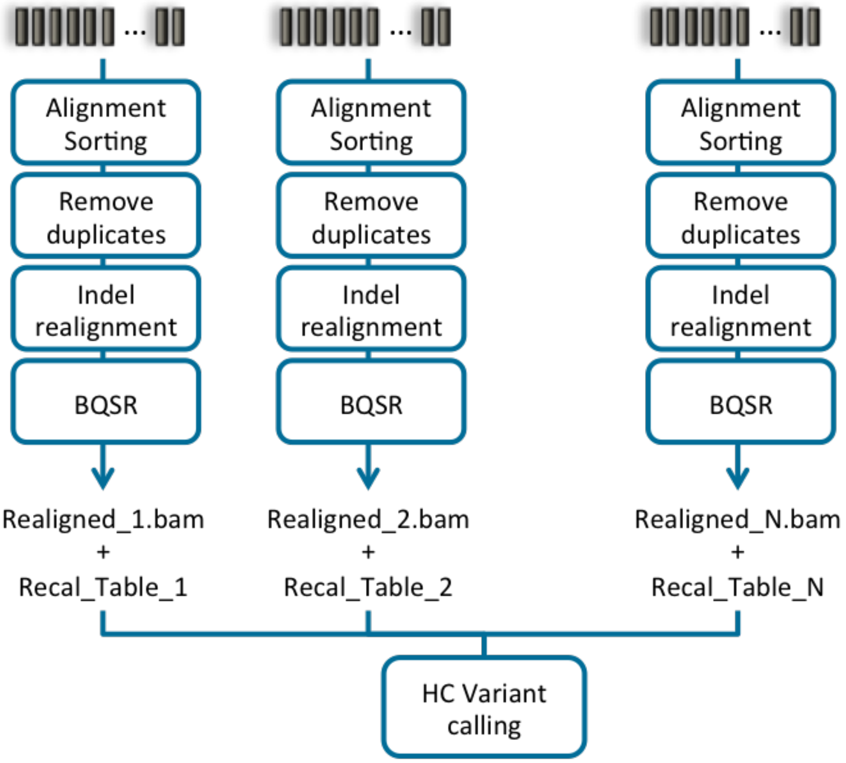


Fig. 9.2: Bioinformatics pipeline for joint calling using Haplotyper and multiple BAM files

(continued from previous page)

```
    --sam2bam -i -
  sentieon driver -t NUMBER_THREADS -i ${sample}_SORTED_BAM \
    --algo LocusCollector --fun score_info ${sample}_SCORE_TXT
  sentieon driver -t NUMBER_THREADS -i SORTED_BAM_1 -i SORTED_BAM_2 \
    --algo Dedup --rmdup --score_info ${sample}_SCORE_TXT ${sample}_DEDUP_BAM
  sentieon driver -r REFERENCE -t NUMBER_THREADS -i ${sample}_DEDUP_BAM \
    --algo Realigner -k $known_sites ${sample}_REALIGNED_BAM
  sentieon driver -r REFERENCE -t NUMBER_THREADS -i ${sample}_REALIGNED_BAM \
    --algo QualCal -k $known_sites ${sample}_RECAL_DATA_TABLE
done
#Run joint variant calling
sentieon driver -t NUMBER_THREADS -i s1_REALIGN_BAM -q s1_RECAL_DATA_TABLE \
  -i s2_REALIGN_BAM -q s2_RECAL_DATA_TABLE -i s3_REALIGN_BAM \
  -q s3_RECAL_DATA_TABLE --algo Haplotyper VARIANT_VCF
```

- Process each sample individually and use the Haplotyper algorithm with option `--emit_mode gvcf` to create a GVCF file containing additional information, then process all GVCF using the GVCFtyper algorithm. This method allows for easy and fast reprocessing after additional samples have been processed.

```
#Process all samples independently until GVCF
for sample in s1 s2 s3; do
  (sentieon bwa mem -M -R '@RG\tID:GROUP_${sample}\tSM:${sample}\tPL:PLATFORM' \
    -t NUMBER_THREADS REFERENCE ${sample}_FASTQ_1 ${sample}_FASTQ_2 || echo -n \
    'error') | sentieon util sort -o ${sample}_SORTED_BAM -t NUMBER_THREADS \
    --sam2bam -i -
```

(continues on next page)

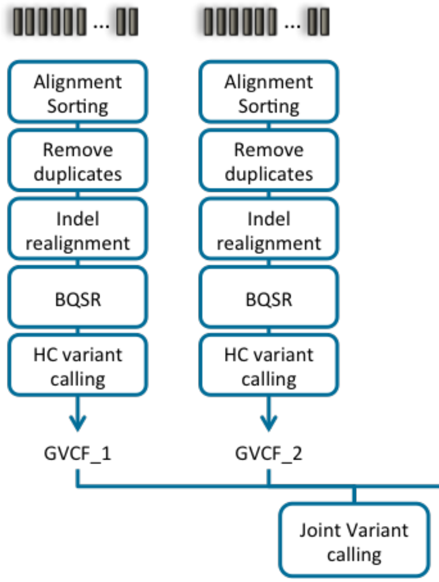


Fig. 9.3: Bioinformatics pipeline for joint calling using Haplotyper in GVCF emit mode and GVCFtyper

(continued from previous page)

```
  sentieon driver -t NUMBER_THREADS -i ${sample}_SORTED_BAM \
    --algo LocusCollector --fun score_info ${sample}_SCORE_TXT
  sentieon driver -t NUMBER_THREADS -i SORTED_BAM_1 -i SORTED_BAM_2 \
    --algo Dedup --rmdup --score_info ${sample}_SCORE_TXT ${sample}_DEDUP_BAM
  sentieon driver -r REFERENCE -t NUMBER_THREADS -i ${sample}_DEDUP_BAM \
    --algo Realigner -k $known_sites ${sample}_REALIGNED_BAM
  sentieon driver -r REFERENCE -t NUMBER_THREADS -i ${sample}_REALIGNED_BAM \
    --algo QualCal -k $known_sites ${sample}_RECAL_DATA_TABLE
  sentieon driver -r REFERENCE -t NUMBER_THREADS -i ${sample}_REALIGN_BAM \
    -q ${sample}_RECAL_DATA_TABLE --algo Haplotyper --emit_mode gvcf \
    ${sample}_VARIANT_GVCF
done
#Run joint variant calling
sentieon driver -r REFERENCE --algo GVCFtyper -v s1_VARIANT_GVCF \
  -v s2_VARIANT_GVCF -v s3_VARIANT_GVCF VARIANT_VCF
```

9.4 Supported annotations in Haplotyper and Genotyper

Both Haplotyper and Genotyper germline variant calling methods accept the option `--annotation` to output additional annotations to the result VCF. The following are the possible annotations supported:

- AC: Allele count in genotypes, for each ALT allele, in the same order as listed
- AF: Allele frequency, for each ALT allele, in the same order as listed
- AN: Total number of alleles in called genotypes
- BaseQRankSum: Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities
- ClippingRankSum: Z-score From Wilcoxon rank sum test of Alt vs. Ref number of hard clipped bases
- DP: Combined depth across samples

- ExcessHet: Phred-scaled p-value for exact test of excess heterozygosity
- FS: Phred-scaled p-value using Fisher's exact test to detect strand bias
- InbreedingCoeff: Inbreeding coefficient as estimated from the genotype likelihoods per-sample when compared against the Hardy-Weinberg expectation
- MLEAC: Maximum likelihood expectation (MLE) for the allele counts, for each ALT allele, in the same order as listed
- MLEAF: Maximum likelihood expectation (MLE) for the allele frequency, for each ALT allele, in the same order as listed
- MQ: RMS mapping quality
- MQ0: Number of MAPQ == 0 reads
- MQRankSum: Z-score From Wilcoxon rank sum test of Alt vs. Ref read mapping qualities
- QD: Variant Confidence/Quality by Depth
- RAW_MQ: Raw data for RMS mapping quality
- ReadPosRankSum: Z-score from Wilcoxon rank sum test of Alt vs. Ref read position bias
- SOR: Symmetric Odds Ratio of 2x2 contingency table to detect strand bias
- SAC: Number of reads on the forward and reverse strand supporting each allele (including reference)
- AS_BaseQRankSum: Allele specific
- AS_FS: Allele specific Phred-scaled p-value using Fisher's exact test to detect strand bias
- AS_InbreedingCoeff: Allele specific Inbreeding coefficient as estimated from the genotype likelihoods per-sample when compared against the Hardy-Weinberg expectation
- AS_MQRankSum: Allele specific Z-score From Wilcoxon rank sum test of Alt vs. Ref read mapping qualities
- AS_QD: Allele specific Variant Confidence/Quality by Depth
- AS_MQ: Allele specific RMS mapping quality
- AS_ReadPosRankSum: Allele specific Z-score from Wilcoxon rank sum test of Alt vs. Ref read position bias
- AS_SOR: Allele specific Symmetric Odds Ratio of 2x2 contingency table to detect strand bias

9.5 Running the license server (LICSRVR) as a system service

If your system follows the Linux Standard Base Core Specifications[4] you can setup the license server to be automatically started in your system by running the following commands as root:

1. Install the license server startup script into the /etc/init.d directory. The startup script is included with the release package.

```
install -m 0755 SENTIEON_DIR/doc/licsrvr.sh /etc/init.d/licsrvr
```

2. Create and customize the configuration file; the configuration file is typically /etc/sysconfig/licsrvr; however, in Ubuntu the configuration file will be /etc/default/licsrvr.

[4] http://refspecs.linuxfoundation.org/LSB_3.1.0/LSB-Core-generic/LSB-Core-generic/tocsysinit.html

```
% cat /etc/sysconfig/licsrvr
```

Below is an example of the configuration file, where $SENTIEON_DIR is the folder where the Sentieon Genomics software release package is located and $LICENSE_DIR is the folder where the license file is located; the license server log file will be stored in /var/log/sentieon_licsrvr.log.

```
licsrvr="SENTIEON_DIR/bin/sentieon licsrvr"
licfile=LICENSE_DIR/LICENSE_FILE.lic
logfile=/var/log/sentieon_licsrvr.log
```

3. Execute the system init script installation script.

   ```
   /usr/lib/lsb/install_initd /etc/init.d/licsrvr
   ```

4. You can then start/stop/restart/check status of the service by using:

   ```
   service licsrvr [start|stop|restart|status]
   ```

For Ubuntu systems, if you do not have the lsb/install_initd binary and choose not to install the lsb-core package, you do not need to run steps 3 and 4, and can run the following commands to install licsrvr as a service:

```
sudo update-rc.d licsrvr defaults
sudo update-rc.d licsrvr enable
```

For Debian systems, if you do not have the lsb/install_initd binary and choose not to install the lsb-core package, you do not need to run steps 3 and 4, and can run the following commands to install licsrvr as a service:

```
#install the service
cd /etc/rc3.d
sudo ln -s ../init.d/licsrvr S05licsrvr

#start the service
/etc/init.d/licsrvr start

#stop the service
/etc/init.d/licsrvr stop

#disable the service
rm /etc/rc3.d/licsrvr
```

9.6 Removing reads after alignment with low mapping quality

Below is an example of how to remove reads with low mapping quality after alignment, so that they do not take space in the output BAM file.

```
sentieon bwa mem -M -R '@RG\tID:GROUP_NAME_1 \tSM:SAMPLE_NAME \tPL:PLATFORM' \
  -t NUMBER_THREADS REFERENCE SAMPLE_1 [SAMPLE_2] \
  |samtools view -h -q MAP_QUALITY_THRESHOLD - \
  | sentieon util sort -o SORTED_BAM_1 -t NUMBER_THREADS --sam2bam -i -
```

9.7 Performing Dedup to mark primary and non-primary reads

The standard 2 pass dedup will ignore non-primary reads and never change their duplicate flag.

Below is an example of how to perform dedup to mark both primary and non-primary reads:

```
sentieon driver -t NUMBER_THREADS -i SORTED_BAM \
  --algo LocusCollector --fun score_info SCORE_TXT
sentieon driver -t NUMBER_THREADS -i SORTED_BAM \
  --algo Dedup --score_info SCORE_TXT --metrics DEDUP_METRIC_TXT \
  -- output_dup_read_name DUPLICATED_READNAMES_TXT
sentieon driver -t NUMBER_THREADS -i SORTED_BAM \
  --algo Dedup --rmdup --dup_read_name DUPLICATED_READNAMES_TXT DEDUP_BAM
```

9.8 Pipeline modifications when using data with quantized quality scores

When your input data has quantized base quality scores, there is a high chance that the base quality reported in the input FASTQ is much higher than the empirical base quality; in this case, it is possible that the variant calling step may take longer to process the data, as low quality bases that would not have been taken into account in the local assembly will cause the generation of many irrelevant low-quality haplotypes. This issue should be addressed by running the Base Quality Score Recalibration step of the pipeline; alternatively, you can modify the default value of the min_base_qual argument of Haplotyper, or TNhaplotyper to reduce the impact: add the option `--min_base_qual 15` to your command line to prevent this issue.

9.9 Modify RG information on BAM files when both Tumor and Normal inputs have the same RGID

If you input multiple different BAM files containing readgroups with the same ID but different attributes, the Sentieon tools will error out. This can happen when in TNseq and TNscope the tumor and normal sample BAM files have an RG ID of “1”. In order to work around this, you can use the `--replace_rg` functionality to modify the RG information.

```
#first lane tumor, RGID incorrectly set to uninformative 1
sentieon bwa mem -M -R '@RG\tID:1\tSM:TUMOR_SM\tPL:PLATFORM' -t NT \
  REFERENCE TUMOR_1_1.fq.gz TUMOR_1_2.fq.gz | sentieon util sort \
  -o TUMOR_1.bam -t NT --sam2bam -i -

#second lane tumor, RGID incorrectly set to uninformative 1
sentieon bwa mem -M -R '@RG\tID:1\tSM:TUMOR_SM\tPL:PLATFORM' -t NT \
  REFERENCE TUMOR_2_1.fq.gz TUMOR_2_2.fq.gz | sentieon util sort \
  -o TUMOR_2.bam -t NT --sam2bam -i -

#normal, RGID incorrectly set to uninformative 1
sentieon bwa mem -M -R '@RG\tID:1\tSM:NORMAL_SM\tPL:PLATFORM' -t NT \
  REFERENCE TUMOR_1.fq.gz TUMOR_2.fq.gz | sentieon util sort \
  -o NORMAL.bam -t NT --sam2bam -i -

#using replace_rg argument to modify the RGID and make them unique
#each replace_RG argument will only affect the following input BAM file
sentieon driver -r REFERENCE \
  --replace_rg 1='ID:T_1\tSM:TUMOR\tPL:PLATFORM' -i TUMOR_1.bam \
```

(continues on next page)

(continued from previous page)

```
  --replace_rg 1='ID:T_2\tSM:TUMOR\tPL:PLATFORM' -i TUMOR_2.bam \
  --replace_rg 1='ID:N\tSM:NORMAL\tPL:PLATFORM'-i NORMAL.bam \
  --algo TNscope --tumor_sample TUMOR --normal_sample NORMAL OUT_TN_VCF
```

10 Troubleshooting

10.1 Preparing reference file for use

If your reference FASTA file has not been pre-processed such that the data specified in Table 3.1 is not available to the software, you will need to process it as explained in https://www.broadinstitute.org/gatk/guide/article?id=2798.

You will need to do the following steps:

1. Generate the BWA index using BWA. This will create the ".fasta.amb", ".fasta.ann", ".fasta.bwt", ".fasta.pac" and ".fasta.sa" files.

```
sentieon bwa index reference.fasta
```

2. Generate the FASTA file index using samtools. This will create the ".fasta.fai" file.

```
samtools faidx reference.fasta
```

3. Generate the sequence dictionary using Picard. This will create the ".dict" file.

```
java -jar CreateSequenceDictionary.jar REFERENCE=reference.fasta \
  OUTPUT=reference.dict
```

10.2 Preparing RefSeq file for use

RefSeq files are used to aggregate the results of the CoverageMetrics algorithm to the gene level.

In order to use RefSeq files downloaded from the ucsc genome browser, they need to be sorted by chromosome and loci. To perform the sorting you will need to do the following steps:

1. Strip the header from the file

```
grep -v "^#" FILE.refSeq > FILE.refSeq.headerless
grep -e "^#" FILE.refSeq > FILE.refSeq.header
```

2. Sort the loci first using unix sort.

```
sort -k 5 -n FILE.refSeq.headerless > FILE.refSeq.presorted
```

3. Use GATK sortByRef.pl using the FASTA index fai to sort by chromosome.

```
perl sortByRef.pl --k 3 FILE.refSeq.presorted FASTA.fai --tmp ~/tmp \
    > FILE_sorted_headerless.refSeq
```

4. Put the header back to the file.

```
cat FILE.refSeq.header > FILE_sorted.refSeq
cat FILE_sorted_headerless.refSeq >> FILE_sorted.refSeq
```

10.3 Common usage problems

Following is a list of symptoms of common problems as well as solutions for them.

10.3.1 Driver or Util fails with Error: can not open file (xxx) in mode(r), Too many open files

The root cause for this error is that the limit of concurrently open files is not set to be high enough for your system.

You can solve this error by setting the system ulimit -n. In a Linux based system:

1. Check the limit of maximum number of open files in your system, by running the following command:

```
ulimit -n
```

2. Set a higher limit by editing file `/etc/security/limits.conf` as root, and add the following 2 lines:

```
* soft nofile 16384
* hard nofile 16384
```

3. If your system is running Ubuntu, you also need to add this line to you shell profile `~/.bashrc`

```
ulimit -n 16384
```

4. You need to log out of your system and log back in for the changes to take effect. After logging in, check that the change was applied correctly by running the following command:

```
ulimit -n
```

5. The command should return 16384.

In a Mac OSX based system:

1. Check the limit of maximum number of open files in your system, by running the following command:

```
ulimit -n
```

2. Set a higher limit by editing or create file `/etc/sysctl.conf`, and add the following 2 lines:

```
kern.maxfilesperproc=16384
kern.maxfiles=16384
```

3. and add this line to you shell profile `~/.bash_profile`

```
ulimit -n 16384
```

4. You need to reboot your system for the changes to take effect. After rebooting, check that the change was applied correctly by running the following command:

```
ulimit -n
```

5. The command should return 16384.

10.3.2 Driver fails with error: Contig XXX from vcf/bam is not present in the reference, or error Contig XXX has different size in vcf/bam than in the reference

The root cause for this error is that the input VCF or BAM file is incompatible with the reference fasta file. Either there are contigs in the file not present in the reference, or the contigs have different sizes. This is most likely caused by using VCF or BAM files processed with a different reference.

10.3.3 Driver reports warning: Contigs in the vcf file XXX do not match any contigs in the reference

The root cause for this warning is that the input VCF file is incompatible with the reference fasta file, and the contigs in the file are not present in the reference. This is most likely caused by using VCF files from a different reference.

10.3.4 License message: No more license available for Sentieon…

This message is produced when you request to run the Sentieon software in more threads than your license currently allows you to. This happens because you are concurrently running commands that collectively request more threads than the number of cores your license supports.

The Sentieon commands will be idle while waiting for free licenses, but the commands will not fail.

10.3.5 Driver fails with error: Readgroup XX is present in multiple BAM files with different attributes

This error is produced when you input two different BAM files containing readgroups with the same ID but different attributes, for instance when in TNseq and Tnscope the tumor and normal sample BAM files have an RG ID of “1”.

Before you are able to use the BAM files you will need to edit them to make the RG ID unique, for instance by adding the SM name to the RG ID. You can check Section 9.9 for an example of a work-around for this issue.

Alternatively, you can use the samtools addreplacerg functionality to do modify the RG ID of the input BAM files and make them unique:

```
#add the new RG and modify all reads in the BAM file
RGtag=$(samtools view -H $INPUT_BAM|grep ^@RG|sed "s|ID:$ORIG_RGID|ID:$NEW_RGID|g")
samtools addreplacerg -r"$RGtag"-o $TMP_BAM $INPUT_BAM
#reheader the BAM to remove the original RG that is no longer used
samtools view -H $TMP_BAM|grep -v"^@RG.*$ORIGINAL_RGID" \
```

(continues on next page)

(continued from previous page)

```
    |$samtools reheader - $TMP_BAM > $OUTPUT_BAM
rm $TMP_BAM
```

10.3.6 Driver reports warning: none of the QualCal tables is applicable to the input BAM files

This warning means that none of the information in the recalibration table input file can be applied to the input BAM file, which is likley due to using a recalibration table that does not correspond to the BAM file.

This warning could be produced when the input BAM file to QualCal does not have the correct fields in the RG fields. For instance, this could happen if the PL tag of the RG is set to something different than ILLUMINA; in this case, you will need to modify the BAM header to include/modify the missing/incorrect fields, for which you can use the samtools reheader functionality.

10.3.7 BWA uses an abnormal amount of memory when using FASTQ files created from a BAM file

When you use FASTQ files created by converting an already sorted BAM file, it may happen that all the unmapped reads are grouped together at the end of the FASTQ inputs. In that case, BWA may use an abnormal amount of memory at the end of the alignment because poorly mapped or unmappable reads require additional memory.

In order to reduce the abnormal memory usage, you should first re-sort the bam file to make sure that the unmapped reads are not grouped together. You can use *samtools* to do that:

```
samtools sort -n -@ 32 input.bam | samtools fastq -@ 32 \
-s >(gzip -c > single.fastq.gz) -0 >(gzip -c > unpaired.fastq.gz) \
-1 >(gzip -c > output_1.fastq.gz) -2 >(gzip -c > output_2.fastq.gz) -
```

10.4 KPNS - Known Problems No Solutions

10.4.1 Lack of support for gzipped vcf files not compressed with bgzip

Normal gzipped files do not allow for random or indexed access to the information contained on them, only files compressed with bgzip are indexable. As such, the Sentieon software does not support gzipped VCF files as input. In order to use these files you will need to uncompress them using gunzip and either use them uncompressed or recompress them with bgzip. Alternativelly, you can use `util vcfconvert` to recompress and index the files.

```
gzip -dc INPUT.vcf.gz | sentieon util vcfconvert - OUTPUT.vcf.gz
```

10.4.2 Lack of support for gzipped fasta files

Currently the software does not support gzipped FASTA files as input. You need to gunzip the files before using them.

10.4.3 FASTQ files required to have SANGER quality format

If your FASTQ files have been encoded with Illumina^{TM} sequencing technology before 1.8, the read quality scores will not be in SANGER format, which may produce unexpected results. The Sentieon Genomics software will not detect that you are using the unsupported format.

10.4.4 Driver fails with error: ImportError: No module named argparse

This error is produced when running tnhapfilter in an environment where the python version is 2.6.x and the argparse module is not present. You will need to install the argparse module to your python installation; you can do this by running `pip install argparse` or whichever other package manager you use.

11 Release notes and usage changes

11.1 Updates from previous releases

11.1.1 Release 201808.01

Type	Description
Feature	Improved performance of DNAscope and Machine learning model
Feature	Improved speed of Insert Size plotting when the sample has very large insert sizes
Bug-fix	Solved issue in GVCFtyper that could create a non conforming VCF when doing joint calling of DNAscope results.
Bug-fix	Solved issue in DNAModelApply that would cause an error when the input file does not contain a ModelID.
Bug-fix	Solved issue in Haplotyper when using the –given option that would generate a VCF without the LowQual filter definition in the header.
Bug-fix	Solved issue in tnhapfilter tool to include tumor sample in the header of the VCF

11.1.2 Release 201808

Type	Description
Feature	Added support in ReadWriter for customized flag read filters.
Feature	Added support to modify the read group information of an input BAM file.
Feature	Added support in TNscope and TNhaplotyper to output a BAM including local reassembly of the reads.
Feature	Added support in TNModelApply to include command line to the VCF output.
Feature	Added support to QualCalFilter to keep the base quality scores before recalibration.
Feature	Introduced TNhaplotyper2 algorithm
Feature	Added support for outputting CRAM v 3.0 files.
Feature	Added support in Haplotyper, DNAscope and GVCFtyper to input the expected heterozygosity value used to compute prior likelihoods.
Feature	Updated interface of plot to better match other tools.
Feature	Added support for machine learning model for DNAscope
Feature	Modified the help for driver options to make it more user friendly.
Feature	Improved speed in Dedup.
Feature	Added support in Haplotyper, Genotyper and DNAscope to include @PG line in the output header of the `--bam_output` argument.
Feature	Updated the @PG information when processing in distribution mode to make it more informative.
Bug-fix	Solved issue in GVCFtyper on DNAscope GVCFs that may report incorrect phasing on a multi-allelic variant
Bug-fix	Solved issue in util sort that would cause an error in older CPUs.
Bug-fix	Added additional error checks and error reporting.
Bug-fix	Solved issue in LocusCollector that would cause a crash when running 2 commands in parallel outputting to the same file.
Bug-fix	Solved issue in Haplotyper that could create a GVCF with REF allele incorrectly set to N under very rare conditions.
Bug-fix	Solved issue in TNscope that may use uninitialized values
Bug-fix	Solved issue in AlignmentStat that would report an incorrect command line in the header of the output file.
Bug-fix	Solved issue that could slow down the startup when the LIBDIR contains many files and is stored in a NFS.

11.1.3 Release 201711.05

Type	Description
Bug-fix	Solved issue in in Haplotyper that could create a GVCF file that does not cover the entire region under rare circumstances.

11.1.4 Release 201711.04

Type	Description
Feature	Added support to all BAM writing tools to preserve existing PG tags in the header.
Bug-fix	Solved issue in TNscope that could cause an error with references containing small contigs.
Bug-fix	Solved issue in SVSolver that could cause a segmentation fault.
Bug-fix	Solved issue in Haplotyper when using the —bam_output option that could generate a debug BAM file incompatible with other tools.
Bug-fix	Solved issue in ApplyVarCal that would generate a non-compliant list of FILTERs if the input file already had existing filters.

11.1.5 Release 201711.03

11.1.6 Release 201711.02

Type	Description
Feature	Added support for BWA shm mode.
Feature	Added QC metrics tools: BaseDistributionByCycle, QualityYield, WgsMetricsAlgo, SequenceArtifactMetricsAlgo.
Feature	Improved speed in TNscope when calling SVs.
Feature	Added support in TNscope to show progress report during the SV calling.
Feature	Reduced memory utilization in TNscope while calling SVs.
Feature	Introduced β-version of Windows based tools.
Feature	Improved speed for the Windows version.
Feature	Added additional error checks and error reporting.
Feature	Modified the default behavior in DNAscope when only calling short variants to filter out chimeric reads during genotyping.
Feature	Reduced memory usage in GVCFtyper.
Bug-fix	Solved issue in Haplotyper that would cause an error if the input BAM file contained reads where the cigar is composed solely of soft and hard clips.
Bug-fix	Solved issue in TNscope that could cause a segmentation fault when using an incomplete BAM file.
Bug-fix	Solved issue in TNscope that would cause a segmentation fault under rare circumstances.
Bug-fix	Solved issue in Haplotyper and DNAscope that could cause an error when using the `bam_output` option.
Bug-fix	Solved issue in Realign that would not do the proper read pairing when dealing with secondary alignments.
Bug-fix	Maintenance update on the SVsolver algorithm.
Bug-fix	Solved issue in licsrvrv for Windows that would prevent it from working with HTTP proxy servers with authentication.
Bug-fix	Solved issue in GVCFtyper that would produce VCF records with non-conformant MQ0 when the input GVCFs include MQ0 annotation.

11.1.7 Release 201711.01

11.1.8 Release 201711

Type	Description
Feature	Added support in Haplotyper to output a BAM including the local reassembly of the reads.
Feature	Introduced β-version of new functionality for applying a Machine Learning model to help with variant filtration in TNscope.
Feature	Introduced β-version of new functionality for Python based recipe creation to drive the Sentieon tools.
Feature	Introduced β-version of new product DNAscope for germline variant calling and germline structural variant calling.
Feature	Added support for printing the command line to stderr log, usually as the first line, before the license check.
Feature	Added support in QualCal to calculate recalibration tables based on PU if present.
Feature	Added support in ApplyVarCal to use both SNP and INDEL models in a single command line.
Feature	Added additional error checks and error reporting.
Bug-fix	Solved issue in TNscope SV that would cause an error if the input BAM file contained realigned reads with no matching bases.
Bug-fix	Solved issue that would cause a crash when using a BAM file containing reads with invalid mate tid.
Bug-fix	Solved issue in GVCFtyper that could cause a job to incorrectly think there are not enough licenses available when the communication to the license server is slow.
Bug-fix	Solved issue in TNscope that failed to detect long INDELs under rare circumstances.
Bug-fix	Solved issue in ApplyVarCal that caused the output VCF file to miss the SentieonCommandLine in the header.
Bug-fix	Solved issue in TNscope that prevented modifying the max_normalAlt_active option, which could cause loss of variants in areas of extreme depth.
Bug-fix	Solved issue in GCBias that would report results for a RG that did not contain any reads.
Bug-fix	Solved issue in util that would cause a crash when the BAM contained reads whose tid is out of bound.
Bug-fix	Solved issue in util stream that would not include the full path of util binary in the PG line of the BAM header.
Bug-fix	Solved issue in AlignmentStat that would produce an inaccurate PCT_ADAPTER value when the adapter_seq is not null.
Bug-fix	Solved issue in ApplyVarCal that would produce incorrect results when using sensitivity 0.
Bug-fix	Solved issue in the help of TNhaplotyper that would misrepresent the default pcr_indel_model.
Bug-fix	Solved issue in BWA that prevented it from working on certain older AMD cpus.
Bug-fix	Solved issue in InsertSizeMetrics that would generate a header using space instead of tab.

11.1.9 Release 201704.03

Type	Description
Feature	Added support for using a HTTP proxy with authentication.
Bug-fix	Solved issue that would prevent recalibration tables from older releases from being applied when the BAM file RGs have a defined PU.
Bug-fix	Solved issue in Realigner that could cause an error under extremely rare circumstances when reads in the input file have inconsistent information.
Bug-fix	Solved issue in TNscope that could incorrectly error out when the tumor_contamination_frac parameter was set to 0.
Bug-fix	Solved issue that was preventing using BAM files with Contigs larger than 300M.

11.1.10 Release 201704.02

11.1.11 Release 201704.01

Type	Description
Feature	Added additional checks on the input BAM files.
Bug-fix	Solved issue in Dedup that would cause the algorithm to hang when the BAM file has more than 4 billion reads (200x WGS with 150 BP reads).
Bug-fix	Solved issue in QualCal that was slowing down the calculation.
Bug-fix	Solved issue in TNscope that could report the wrong reference allele for structural variants.
Bug-fix	Solved issue that could cause a job to hang under extremely rare circumstances.
Bug-fix	Solved issue that could cause extra tags in the BAM header to disappear.

11.1.12 Release 201704

Type	Description
Feature	Maintenance update of algorithms.
Feature	Added support for defining a temporary directory, instead of using PWD.
Feature	Added support for type 2 VCF index files.
Feature	Added additional error checks and error reporting.
Feature	Added support for SAC annotation and Allele Specific annotations.
Feature	Added support for additional read filtering: MapQualFilter and OverclippingFilter.
Feature	Added support for CollectVCMetrics.
Feature	Added support for Deduplication equivalent to using BAM files sorted by read name.
Bug-fix	Solved issue in license control that would prevent run when the SENTIEON_AUTH_DATA is of a specific length.
Bug-fix	Solved issue that would not properly parse bed files containing both space and tab delimited fields.
Bug-fix	Solved issue in TNscope that would report the wrong REF allele for structural variants.
Bug-fix	Solved issue in CoverageMetrics that would produce the wrong summary mean coverage when not using a bed file.

11.1.13 Release 201611.03

Type	Description
Feature	Added support for more options for base quality score recalibration.
Bug-fix	Solved issue in GVCFtyper that could report an incorrect GT or PL when the input GVCFs had been processed with different bed files.
Bug-fix	Solved issue in Realign that caused reads close to the edge of the contig to be realigned beyond the contig boundary.
Bug-fix	Solved issue in Realign that caused a segmentation fault when using known sites VCF including symbolic variants.
Bug-fix	Added support for Realign to update NM and MD tags in the realigned reads.

11.1.14 Release 201611.02

Type	Description
Feature	Added support for calling given known variants to Genotyper, Haplotyper, and TNscope.
Feature	Added support for outputting physical phasing information of variants in TNhaplotyper and TNscope.
Feature	Added support for base quality correction in streaming mode in util.
Feature	Maintenance update of HsMetricAlgo.
Feature	Removed redundant nthr argument in VarCal.
Bug-fix	Solved issue in TNscope causing excessive memory utilization in the structural variant calling.
Bug-fix	Solved issue in TNscope causing a segmentation fault when an output file was not set.
Bug-fix	Solved issue in QualCal in `--plot` mode that was incorrectly reporting the need for known sites.
Bug-fix	Solved incorrect description of filter low_t_alt_frac in TNscope.

11.1.15 Release 201611.01

Type	Description
Feature	Removed TNscope temporary files novo_hap.data and novo_sv.data.
Feature	Increased default license timeout from client request.
Bug-fix	Solved an issue causing jobs to hang at the end of the processing when there were issues in the network communication.
Bug-fix	Solved an issue in TNscope causing an assertion error when pairend reads have inconsistent flags.
Bug-fix	Solved an issue in TNscope that incorrectly considered reads marked as duplicates in the calculation.

11.1.16 Release 201611

Type	Description
Feature	Added `--emit_mode all` in GVCFtyper.
Feature	Updates to β-version of TNscope.
Feature	Speed improvement to the alignment and sorting tools.
Feature	Added ContaminationAssessment algorithm for contamination estimation.
Feature	Added detection of truncated VCF input and corrupted cram input files.
Bug-fix	Solved an issue in VarCal that was not producing a plot file when the recalibration failed.
Bug-fix	Solved an issue in VarCal that could cause a segmentation fault when there were not enough licenses for the requested threads.
Bug-fix	Solved an issue in util sort that was causing a segmentation fault when converting an empty sam file to BAM file.

11.1.17 Release 201608.01

Type	Description
Bug-fix	Solved issue in TNseq in tumor-only mode that added to the output VCF an empty column for non-existent NORMAL sample.

11.1.18 Release 201608

Type	Description
Feature	Introduced β-version of new product TNscope for Tumor-Normal somatic variant calling and structural variant calling.
Feature	Reduced peak memory utilization in klib for alignment.
Feature	Speed improvement in the input file loading of GVCFtyper.
Feature	Speed improvement in the Haplotyper algorithm.
Feature	Made VarCal more robust when building the VQSR gaussian models.
Bug-fix	Cleared up certain error messages to make them more user friendly.
Bug-fix	Solved issue in Genotyper that caused a segmentation fault when using var_type BOTH.

11.1.19 Release 201606.02

Type	Description
Feature	Reduced memory utilization in GVCFtyper to reduce requirements for analysis of large (4000+) cohorts.
Bug-fix	Solved issue in TNsnv that created VCF files non-conforming to the standard.

11.1.20 Release 201606.01

Type	Description
Bug-fix	Solved an issue that prevented licsrvr from serving licenses when the user group list is longer than 1000 characters.

11.1.21 Release 201606

Type	Description
Feature	Maintenance update of TNhaplotyper algorithm (still in β).
Feature	Added support for RNAseq variant calling.
Feature	Added online help.
Feature	Added support for interval padding.
Bug-fix	Solved issue that produced no output in TNhaplotyper when using a PoN or no normal sample data.
Bug-fix	Solved issue preventing BED file with a header from being used.

11.1.22 Release 201603.03

Type	Description
Feature	Added method for inputting long list of VCF files to GVCFtyper.
Feature	Added command line to VCF header.
Feature	Added support for additional annotations on all variant callers.
Bug-fix	Solved issue producing an assertion error when too few variants are called.
Bug-fix	Solved issue reporting the wrong assembly in the VCF header when the FASTA file cannot be interpreted correctly.

11.1.23 Release 201603.02

Type	Description
Bug-fix	Solved issue in TNhaplotyper that incorrectly filtered INDELs.
Bug-fix	Solved issue in TNsnv and TNhaplotyper that made the dbsnp a required argument.
Bug-fix	Solved issue that creates non-conforming VCF files when calling variants in locations where the REF base is M.

11.1.24 Release 201603.01

Type	Description
Feature	Speed improvement in the VQSR algorithm.
Bug-fix	Solved issue that caused VQSR to apply to the wrong type of variants when the variants had a prior LOD annotation.
Bug-fix	Solved issue that slowed down dedup optical duplicate calculation when the number of duplicates at one locus is too large.

11.1.25 Release 201603

Type	Description
Feature	Maintenance update of algorithms.
Feature	Added TNsnv and TNhaplotyper (in β) algorithms for Tumor-Normal and Tumor only somatic variant calling.
Feature	Added support for CRAM files.
Feature	Added support for Haploid and Polyploid samples.
Feature	Added support for setting custom GQ bands for GVCF output.
Bug-fix	Solved issue that dropped reads when the reads extended beyond the contig boundary.

11.1.26 Release 201601.01

Type	Description
Bug-fix	Solved issue in GVCFtyper that produced the wrong variant quality on sites with AAF 0.5 when using more than 50 samples.

11.1.27 Release 201601

Type	Description
Feature	Added support for using interval files only on the realign target creation portion of Realigner.
Bug-fix	Solved issue preventing license server from running in systems with an Infiniband interface.
Bug-fix	Solved issue when merging BAM files that marks reads unmapped when their mate is unmapped.
Bug-fix	Solved issue in Haplotyper in GVCF mode when intervals are used that causes missed variants when the first interval of a contig contains no variants
Bug-fix	Solved issue in GVCFtyper to remove stray/empty SB annotations from the output VCF
Bug-fix	Solved issue causing the corruption of the output VCF under rare conditions involving large number of samples.
Bug-fix	Solved issue that prevented using BAM index files without the optional metadata pseudo bin.

11.1.28 Release 201511.01

Type	Description
Bug-fix	Solved issue in QualCal algorithm that slowed down execution when using BAM files with large number of Readgroups.
Bug-fix	Solved issue in ReadWriter algorithm that produced an incorrect BAM file when using a recalibration table and an input BAM that had PGZ auxiliary data.
Bug-fix	Solved issue in VQSR algorithm that caused the execution to hang when using the results from joint calling of more than 20 samples.

11.1.29 Release 201511

Type	Description
Feature	Maintenance update of algorithms.
Feature	Added pcr_indel_model flag to Haplotyper tool.
Feature	Added phasing information to the output of Haplotyper.
Feature	Added depth metrics tool.
Feature	Added support for compressed VCF/GVCF input and output.
Feature	Added support for Picard style interval files.
Feature	Added wrapper script to remove requirement of LD_LIBRARY_PATH environmental library.
Bug-fix	Added FILTER definition to the VQSR output file header.
Bug-fix	Solved issue that would cause an Out of Memory error in BQSR when using interval files.
Bug-fix	Solved issue that would cause a crash in IndelRealigner when the BAM file has reads with supplementary alignment.

11.1.30 Release 201509

Type	Description
Feature	Reduced resource requirements, including reducing the required maximum number of open files in the system.
Feature	Various improvements in error reporting.
Bug-fix	Solved issue that prevented using multiple BAM files as input to algorithms that produced another BAM file.
Bug-fix	Solved issue that prevented running BQSR when using Readgroups containing spaces.
Bug-fix	Solved issue in Haplotyper that could cause an incorrect variant quality score calculation in low coverage regions. The change could be up to 2% in the quality score, for 1 in 100000 variants.

11.1.31 Release 201508.01

Type	Description
Bug-fix	Solved issue in Realign algorithm that may cause an error when using known sites

11.1.32 Release 201508

Type	Description
Feature	Enhanced error reporting and error handling when dealing with inconsistent input data
Feature	Packaged BWA 0.7.12, and removed non-official options for BWA interface. The results of the packaged BWA are identical to the official BWA
Feature	Grouped the temporary files for VQSR plotting into a single file
Feature	New license server for large clusters
Bug-fix	Solved error in Haplotype Caller when using GVCF emit mode for joint calling
Bug-fix	Solved error in joint calling GVCFtyper that prevented using more than 20 samples

11.1.33 Release 201506

Type	Description
Feature	Added support for Hybrid Selection Analysis metrics
Feature	Unified Genotyper default behavior is to call only SNP variants, to be consistent with GATK 3.3 default behavior
α-Feature	Added initial support for joint variant calling of multiple samples that have ben previously processed individually
Bug-fix	Metrics properly reports unmapped reads when both mates in the pair are unmapped

11.1.34 Release 201505.02

Type	Description
Feature	Speed improvement in Alignment stage
Feature	Speed improvement in Dedup stage
Feature	Added support for temporary BAM files

11.1.35 Release 201505.01

Type	Description
Bug-fix	Issue running VQSR in Mac platform

11.1.36 Release 201505

Type	Description
Feature	Speed improvement via AVX optimization
Feature	Standarized option naming convention
Feature	tmp folder moved to the same location as the job
Feature	Added validation checks to input files
Bug-fix	Solved run to run differences in BQSR when reads have quality scores of 0.
Bug-fix	BQSR applied twice when running variant calling using a recaled BAM file.

11.2 Usage changes from previous release

11.2.1 Release 201808.01

This released introduced the following changes in interface and usage:

- The binaries licsrvr and licclnt require using the bin/sentieon wrapper to be expecuted.

11.2.2 Release 201808

This released introduced the following changes in interface and usage:

- Added `--bam_output` option to TNscope and TNhaplotyper to output a BAM file including local reassembly of the reads.
- Added support for running BWA through the sentieon command wrapper.
- Modified plot interface to be consistent with other tools.
- Added support for CRAM v3 files.
- Added options `--snp_heterozygosity` and `--indel_heterozygosity` to Haplotyper and GVCFtyper to input the expected heterozygosity value used to compute prior likelihoods.
- Added options `--keep_oq` and `--use_oq` to QualCalFilter to keep and use the original qualities when performing BQSR.
- Added option `--read_flag_mask` to ReadWriter to perform customized read filtering.
- Added option `--replace_rg` to driver to modify the RG information of an input BAM file.
- Modified the flow for germline variant calling with DNAscope and added DNAModelApply algorithm.

11.2.3 Release 201711.05

There are no changes in interface for this release.

11.2.4 Release 201711.04

There are no changes in interface for this release.

11.2.5 Release 201711.03

This released introduced the following changes in interface and usage:

- Introduced multi threaded capability to BWA shm, so that BWA shm accepts the -t NUMBER_THREADS option.
- Introduced new environmental variable bwt_max_mem to limit the memory usage of BWA at the expense of speed performance.

11.2.6 Release 201711.02

This released introduced the following changes in interface and usage:

- Introduced new QC metrics algorithms: BaseDistributionByCycle, QualityYield, WgsMetricsAlgo, SequenceArtifactMetricsAlgo.
- Modified the default for option filter_chimeric_reads in DNAscope when var_type is NOT bnd.

11.2.7 Release 201711.01

This released introduced the following changes in interface and usage:

- Added options ``--version` and ``--dump`` to licsrvr binary.

11.2.8 Release 201711

This released introduced the following changes in interface and usage:

- Added new DNAscope algorithm for germline variant calling and structural variant calling.
- Added new TNModelApply algorithm for applying a Machine Learning model to help with variant filtration in TNscope.
- Added new `--bam_output` option to Haplotyper to output a BAM file including local reassembly of the reads.
- Added new option `--vqsr_model` to ApplyVarCal to allow application of both SNP and INDEL models in a single command line.

11.2.9 Release 201704.03

There are no changes in interface for this release.

11.2.10 Release 201704.02

- Added `--cycle_val_max` parameter in QualCal algorithm.
- Added `--cram_write_options` in Dedup, Realigner, ReadWriter, and RNASplitReadsAtJunction algorithms when output format is CRAM.
- Removed `--min_iter` in VarCal algorithm.

11.2.11 Release 201704.01

There are no changes in interface for this release.

11.2.12 Release 201704

This released introduced the following changes in interface and usage:

- Added new driver option `--temp_dir` option to override where the temporary files will be stored.
- Added extra annotations for the `--annotation` option: SAC, AS_BaseQRankSum, AS_FS, AS_InbreedingCoeff, AS_MQRankSum, AS_QD, AS_MQ, AS_ReadPosRankSum, AS_SOR
- Added new read filter driver options `--read_filter MapQualFilter` to filter the input BAM reads by mapping Quality and `--read_filter OverclippingFilter` to filter the input BAM reads according to their soft clipping characteristics.
- Added CollectVCMetrics algorithm to calculate metrics on the VCF output from variant calling.
- Added new Dedup options `--output_dup_read_name` and `--dup_read_name` to perform dedup to mark both primary and non-primary reads.
- Added new options for TNhaplotyper
- Added a new ContaminationAssessment option `--population` to specify the population to represent the baseline allele fraction.

11.2.13 Release 201611.03

This released introduced the following changes in interface and usage:

- Added new driver option `--read_filter QualCalFilter` to perform base quality score recalibration with additional options.

11.2.14 Release 201611.02

This released introduced the following changes in interface and usage:

- Introduced `--given` option in Genotyper, Haplotyper, and Tnscope, to perform calling at given variant sites.
- Introduced stream mode in util to perform base quality correction in streaming mode.
- Removed `--nthr` argument from VarCal.
- Added extra options to HsMetricAlgo.

11.2.15 Release 201611.01

There are no changes in interface for this release.

11.2.16 Release 201611

This released introduced the following changes in interface and usage:

- Introduced ContaminationAssessment algorithm to estimate the contamination in the tumor sample based on the normal variants called.
- Added `--emit_mode all` option in GVCFtyper.

11.2.17 Release 201608.01

There are no changes in interface for this release.

11.2.18 Release 201608

This released introduced the following changes in interface and usage:

- Introduced TNscope algorithm as part of the new TNscope product for Tumor-Normal somatic variant calling and structural variant calling.

11.2.19 Release 201606.02

There are no changes in interface for this release.

11.2.20 Release 201606.01

There are no changes in interface for this release.

11.2.21 Release 201606

This released introduced the following changes in interface and usage:

- Added RNA variant calling tool `--algo SplitReadsAtJunction` and Haplotyper option `--trim_soft_clip` to process RNA data aligned using STAR.
- Added `--help` driver option to driver and algorithms to show online help.
- Added `--interval_padding` driver option to pad the input intervals.

11.2.22 Release 201603.03

This released introduced the following changes in interface and usage:

- Added `--annotation` option to Haplotyper and Genotyper to output additional annotations to the result VCF.
- Added new method of inputting GVCF files to GVCFtyper.

11.2.23 Release 201603.02

There are no changes in interface for this release.

11.2.24 Release 201603.01

There are no changes in interface for this release.

11.2.25 Release 201603

This released introduced the following changes in interface and usage:

- Added TNsnv and TNhaplotyper algorithm for Tumor-Normal and Tumor only somatic variant calling.
- Haplotyper and Genotyper have a new optional argument `--ploidy` to indicate the ploidicity of the sample being processed.
- All locations where you can input and output BAM files now accept CRAM files.

11.2.26 Release 201601.01

There are no changes in interface for this release.

11.2.27 Release 201601

This released introduced the following changes in interface and usage:

- Realigner has a new optional argument `--interval_list` to use interval file on the target interval portion of the algorithm, to be in line with Broad's Best Practice recommendations. Using option `--interval` in the driver call for realigner would apply the interval to both target creation and realignment, dropping reads in the output BAM file.

11.2.28 Release 201511.01

There are no changes in interface for this release.

11.2.29 Release 201511

This released introduced the following changes in interface and usage:

- Haplotyper has a new optional argument `--pcr_indel_model` to determine the indel model used to weed out false positive indels.
- The single interval definition for the `--interval` driver option is now first-base-1 based as opposed to first-base-0 based. The `--interval` driver option supports Picard style interval lists.
- The driver and util binaries are now accessible through a wrapper script that takes care of setting the LD_LIBRARY_PATH environmental library.
- Output VCF files can be produced compressed by using .vcf.gz or .g.vcf.gz extensions.

11.2.30 Release 201509

This release introduced the following changes in interface and usage:

- It is no longer necessary to setup a large limit of open files.
- In the Remove duplicates stage, the Dedup command now creates an output file reporting the results of the de-duplication.
- Added option `--bam_compression` to all algorithms producing an output BAM file (Dedup, Realign and ReadWriter). The option controls the output BAM file compression level and can be used to achieve a faster speed at the expense of file size.

11.2.31 Release 201508.01

There are no changes in interface for this release.

11.2.32 Release 201508

This release introduced the following changes in interface and usage:

- The BWA mapping binary no longer requires or accepts the option -f.
- The procedure to generate the plots for base quality score recalibration (BQSR) is now composed of 3 commands instead of 2. Option `--pre` has been replaced by option `--before`, and a new option `--plot` is used to generate the plotting data.
- VQSR VarCal algorithm now produces a single file containing all data required to create the VQSR report; this file is specified via the new option `--plot_file`.

11.2.33 Release 201506

There are no changes in interface for this release.

11.2.34 Release 201505.02

This released introduced the following changes in interface and usage:

- In the Alignment stage, the samtools command to convert the output of BWA from SAM to BAM is no longer required, as it is now done in the util command via a new option `--sam2bam`.
- In QualCal algorithm there is a new option `--pre` RECAL_DATA.TABLE to calculate the data required to create the report.

11.2.35 Release 201505.01

There are no changes in interface for this release.

11.2.36 Release 201505

This release introduced the following changes in interface and usage:

- In the GCBias algorithm, the option -s has been replaced by `--summary`.
- In the LocusCollector algorithm, the option -f has been replaced by `--fun`.
- In the Remove duplicates stage, the option -s has been replaced by `--score_info`.
- In the Remove duplicates stage, the option -r has been replaced by `--rmdup`.
- In the BQSR stage, it is no longer necessary to run the varplot command to calculate the data required to create the BQSR report; this call has been merged with the call to apply the recalibration. New option `--csv` is now used.

12 Acknowledgements

Portions of the SENTIEON SOFTWARE may use the following copyrighted material. We acknowledge these developers for their contributions.

```
Copyright (c) 2008, 2009, 2011 by Attractive Chaos <attractor@live.co.uk>

 Permission is hereby granted, free of charge, to any person obtaining
 a copy of this software and associated documentation files (the
 "Software"), to deal in the Software without restriction, including
 without limitation the rights to use, copy, modify, merge, publish,
 distribute, sublicense, and/or sell copies of the Software, and to
 permit persons to whom the Software is furnished to do so, subject to
 the following conditions:

 The above copyright notice and this permission notice shall be
 included in all copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
 BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
 ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
 CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 SOFTWARE.

Copyright (C) 2008-2014 Genome Research Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

Copyright (c) 2012-2013 Genome Research Ltd.
Author: James Bonfield <jkb@sanger.ac.uk>

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

   1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

   2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

   3. Neither the names Genome Research Ltd and Wellcome Trust Sanger
Institute nor the names of its contributors may be used to endorse or promote
products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY GENOME RESEARCH LTD AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL GENOME RESEARCH LTD OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

* ====================================================================
* Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without
 * prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 * acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ''AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com). This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 /

/ Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 /


Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and
   distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the
   copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other
   entities that control, are controlled by, or are under common control with
   that entity.  For the purposes of this definition, "control" means (i) the
   power, direct or indirect, to cause the direction or management of such
   entity, whether by contract or otherwise, or (ii) ownership of fifty percent
   (50%) or more of the outstanding shares, or (iii) beneficial ownership of
   such entity.

   "You" (or "Your") shall mean an individual or Legal Entity exercising
   permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation source, and
   configuration files.

   "Object" form shall mean any form resulting from mechanical transformation
   or translation of a Source form, including but not limited to compiled
   object code, generated documentation, and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or Object form,
   made available under the License, as indicated by a copyright notice that is
   included in or attached to the work (an example is provided in the Appendix
   below).

   "Derivative Works" shall mean any work, whether in Source or Object form,
   that is based on (or derived from) the Work and for which the editorial
   revisions, annotations, elaborations, or other modifications represent, as a
   whole, an original work of authorship. For the purposes of this License,
   Derivative Works shall not include works that remain separable from, or
   merely link (or bind by name) to the interfaces of, the Work and Derivative
   Works thereof.

   "Contribution" shall mean any work of authorship, including the original
   version of the Work and any modifications or additions to that Work or
   Derivative Works thereof, that is intentionally submitted to Licensor for
   inclusion in the Work by the copyright owner or by an individual or Legal
   Entity authorized to submit on behalf of the copyright owner. For the
   purposes of this definition, "submitted" means any form of electronic,
   verbal, or written communication sent to the Licensor or its
   representatives, including but not limited to communication on electronic
   mailing lists, source code control systems, and issue tracking systems that
   are managed by, or on behalf of, the Licensor for the purpose of discussing
   and improving the Work, but excluding communication that is conspicuously
   marked or otherwise designated in writing by the copyright owner as "Not a
   Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity on
   behalf of whom a Contribution has been received by Licensor and subsequently
   incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this
   License, each Contributor hereby grants to You a perpetual, worldwide,
   non-exclusive, no-charge, royalty-free, irrevocable copyright license to
   reproduce, prepare Derivative Works of, publicly display, publicly perform,
   sublicense, and distribute the Work and such Derivative Works in Source or
   Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
   License, each Contributor hereby grants to You a perpetual, worldwide,
   non-exclusive, no-charge, royalty-free, irrevocable (except as stated in
   this section) patent license to make, have made, use, offer to sell, sell,
   import, and otherwise transfer the Work, where such license applies only to
   those patent claims licensable by such Contributor that are necessarily
   infringed by their Contribution(s) alone or by combination of their
   Contribution(s) with the Work to which such Contribution(s) was submitted.
   If You institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work or a
   Contribution incorporated within the Work constitutes direct or contributory
   patent infringement, then any patent licenses granted to You under this
   License for that Work shall terminate as of the date such litigation is
   filed.

4. Redistribution. You may reproduce and distribute copies of the Work or
   Derivative Works thereof in any medium, with or without modifications, and
   in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a
       copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating
       that You changed the files; and
```

```
    (c) You must retain,in the Source form of any Derivative Works that You
        distribute, all copyright, patent, trademark, and attribution notices
        from the Source form of the Work, excluding those notices that do not
        pertain to any part of the Derivative Works; and

    (d) If the Work includes a "NOTICE" text file as part of its distribution,
        then any Derivative Works that You distribute must include a readable
        copy of the attribution notices contained within such NOTICE file,
        excluding those notices that do not pertain to any part of the
        Derivative Works, in at least one of the following places: within a
        NOTICE text file distributed as part of the Derivative Works; within the
        Source form or documentation, if provided along with the Derivative
        Works; or, within a display generated by the Derivative Works, if and
        wherever such third-party notices normally appear. The contents of the
        NOTICE file are for informational purposes only and do not modify the
        License. You may add Your own attribution notices within Derivative
        Works that You distribute, alongside or as an addendum to the NOTICE
        text from the Work, provided that such additional attribution notices
        cannot be construed as modifying the License.

    You may add Your own copyright statement to Your modifications and may
    provide additional or different license terms and conditions for use,
    reproduction, or distribution of Your modifications, or for any such
    Derivative Works as a whole, provided Your use, reproduction, and
    distribution of the Work otherwise complies with the conditions stated in
    this License.

 5. Submission of Contributions. Unless You explicitly state otherwise, any
    Contribution intentionally submitted for inclusion in the Work by You to the
    Licensor shall be under the terms and conditions of this License, without
    any additional terms or conditions. Notwithstanding the above, nothing
    herein shall supersede or modify the terms of any separate license agreement
    you may have executed with Licensor regarding such Contributions.

 6. Trademarks. This License does not grant permission to use the trade names,
    trademarks, service marks, or product names of the Licensor, except as
    required for reasonable and customary use in describing the origin of the
    Work and reproducing the content of the NOTICE file.

 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in
    writing, Licensor provides the Work (and each Contributor provides its
    Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
    KIND, either express or implied, including, without limitation, any
    warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or
    FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining
    the appropriateness of using or redistributing the Work and assume any risks
    associated with Your exercise of permissions under this License.

 8. Limitation of Liability. In no event and under no legal theory, whether in
    tort (including negligence), contract, or otherwise, unless required by
    applicable law (such as deliberate and grossly negligent acts) or agreed to
    in writing, shall any Contributor be liable to You for damages, including
    any direct, indirect, special, incidental, or consequential damages of any
    character arising as a result of this License or out of the use or inability
```

```
   to use the Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all other
   commercial damages or losses), even if such Contributor has been advised of
   the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or
   Derivative Works thereof, You may choose to offer, and charge a fee for,
   acceptance of support, warranty, indemnity, or other liability obligations
   and/or rights consistent with this License. However, in accepting such
   obligations, You may act only on Your own behalf and on Your sole
   responsibility, not on behalf of any other Contributor, and only if You
   agree to indemnify, defend, and hold each Contributor harmless for any
   liability incurred by, or claims asserted against, such Contributor by
   reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS


/*********************************************************************
The MIT License

Copyright (c) 2014 Intel Corporation

        Permission is hereby granted, free of charge, to any person
        obtaining a copy of this software and associated documentation
        files (the "Software"), to deal in the Software without
        restriction, including without limitation the rights to use,
        copy, modify, merge, publish, distribute, sublicense, and/or
        sell copies of the Software, and to permit persons to whom the
        Software is furnished to do so, subject to the following
        conditions:

        The above copyright notice and this permission notice shall be
        included in all copies or substantial portions of the
        Software.

        THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
        KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
        WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
        PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
        COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
        LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
        OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
        SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
******************************************************************/

  Copyright (c) 2013-2015 Niels Lohmann.

  Permission is hereby granted, free of charge, to any person obtaining a copy
  of this software and associated documentation files (the "Software"), to deal
  in the Software without restriction, including without limitation the rights
  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
  copies of the Software, and to permit persons to whom the Software is
  furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

The MIT License (MIT)

Copyright (c) Microsoft Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

13 Acronyms and Abbreviations

GATK Genome Analysis Tool Kit

VCF Variant Call Format

SAM Sequence Alignment/Map

BAM Binary compressed SAM

BCF Binary compressed variant Call Format

BED Browser Extensible Display

indels INsertion-DELetions

14

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

SENTIEON, INC. RESERVES ALL RIGHTS IN THE PROGRAM AS DELIVERED. THE PROGRAM OR ANY PORTION THEREOF MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER EXCEPT AS PROVIDED BY LICENSE, WITHOUT THE CONSENT OF SENTIEON.

THIS NOTICE MAY NOT BE REMOVED FROM THE PROGRAM.

NEITHER SENTIEON, ANY MEMBER OF SENTIEON, THE ORGANIZATION(S) BELOW, NOR ANY PERSON ACTING ON BEHALF OF ANY OF THEM:

1. MAKES ANY WARRANTY OR REPRESENTATION WHATSOEVER, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS OF ANY PURPOSE WITH RESPECT TO THE PROGRAM; OR
2. ASSUMES ANY LIABILITY WHATSOEVER WITH RESPECT TO ANY USE OF THE PROGRAM OR ANY PORTION THEREOF OR WITH RESPECT TO ANY DAMAGES WHICH MAY RESULT FROM SUCH USE.

TO THE MAXIMUM EXTENT PERMITTED BY LAW AND EXCEPT AS EXPRESSLY WARRANTED HEREIN, THE SENTIEON SOFTWARE, DOCUMENTATION, AND OTHER PRODUCTS AND SERVICES PROVIDED BY SENTIEON HEREUNDER ARE PROVIDED AS-IS WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF TITLE OR IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE. SENTIEON AND ITS LICENSORS DO NOT GUARANTEE THAT THE SENTIEON SOFTWARE, DOCUMENTATION OR SERVICES WILL MEET LICENSEE'S REQUIREMENTS, BE ERROR-FREE, OR OPERATE WITHOUT INTERRUPTION.

Sentieon

465 Fairchild Drive, Suite 135, Mountain View CA 94043
www.sentieon.com