pacbio_germline (Beta)

Table of contents

Run the germline variant tool to generate BAM and variants on long read sequences using minimap2 for alignment as well as the DeepVariant variant caller.

Quick Start

```
# This command assumes all the inputs are in INPUT_DIR and all the outputs go to OUTPUT_DIR. docker run --rm --gpus all --volume INPUT_DIR:/workdir --volume OUTPUT_DIR:/outputdir \ --workdir /workdir \ nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1 \ $ pbrun pacbio_germline \ --ref /workdir/${REFERENCE_FILE} \ --in-fq /workdir/${INPUT_FASTQ} \ --out-bam /outputdir/${OUTPUT_BAM} \ --out-variants /outputdir/${OUTPUT_VCF}
```

Compatible CPU-based minimap2, GATK4, and Google DeepVariant Commands

The commands below are the minimap2-v2.26, GATK4, and Google DeepVariant counterpart of the Clara Parabricks command above. The output from these commands will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
# Run minimap2 and pipe the output to create a sorted BAM. $ minimap2 -ax map-pbmm2 \ <INPUT_DIR>/${REFERENCE_FILE} \ <INPUT_DIR>/${INPUT_FASTQ} | \ gatk SortSam \ --java-options -Xmx30g \ --MAX_RECORDS_IN_RAM 5000000 \ -I /dev/stdin \ -O cpu.bam \ --SORT_ORDER coordinate # Run deepvariant BIN_VERSION="1.6.1" sudo docker run \ -v "${PWD}":"/input" \ -v "${PWD}/output":"/output" \ -v "${PWD}/Ref":"/reference" \ google/deepvariant:"${BIN_VERSION}" \ /opt/deepvariant/bin/run_deepvariant \ --model_type PACBIO \ --ref /reference/${REFERENCE_FILE} \ --reads cpu.bam \ --output_vcf /output/"${OUTPUT_VCF_FILE}" \ --num_shards $(nproc) \ --make_examples_extra_args "ws_use_window_selector_model=true"
```

Please note that two changes must be made to the baseline minimap2 code in order to match the results exactly:

Firstly, a new preset must be made in `options.c` in the `mm_set_opt` function that tries to replicate the preset of pbmm2 by setting these parameters as a new preset named "map-pbmm2":

```
io->k = 19; io->w = 10; io->batch_size = 0x7fffffffffffffffL; // always build a uni-part index mo->flag |= MM_I_HPC; mo->flag |= MM_F_CIGAR; mo->flag |= MM_F_LONG_CIGAR; mo->flag |= MM_F_EQX; mo->flag |= MM_F_SOFTCLIP; mo->flag |= MM_F_NO_PRINT_2ND; mo->flag |= MM_F_HARD_MLEVEL; mo->mask_level = 0; mo->e2 = 1; mo->zdrop = 400; mo->a = 2; mo->b = 5; mo->q = 5; mo->q2 = 56; mo->e = 4; mo->zdrop_inv = 50; mo->bw = 2000;
```

Secondly, a fix must be made to the baseline KSW2 code to round the loop fission start and end points by changing them to `st` and `en` respectively. If the start point (`st0`) is a number below 16, but greater than 0, its scoring values will not be initialized correctly, but will still be used later when computing the actual alignment. This can be fixed by rounding the start and end points to multiples of 16.

To make this fix, change the following code in `ksw2_extd2_sse.c`:

```
// loop fission: set scores first if (!(flag & KSW_EZ_GENERIC_SC)) { for (t = st0; t <= en0; t += 16) { __m128i sq, st, tmp, mask; sq = _mm_loadu_si128((__m128i*)&sf[t]); st = _mm_loadu_si128((__m128i*)&qrr[t]); mask = _mm_or_si128(_mm_cmpeq_epi8(sq, m1_), _mm_cmpeq_epi8(st, m1_)); tmp = _mm_cmpeq_epi8(sq, st); #ifdef __SSE4_1__ tmp = _mm_blendv_epi8(sc_mis_, sc_mch_, tmp); tmp = _mm_blendv_epi8(tmp,
sc_N_, mask); #else tmp = _mm_or_si128(_mm_andnot_si128(tmp, sc_mis_),
_mm_and_si128(tmp, sc_mch_)); tmp = _mm_or_si128(_mm_andnot_si128(mask,
tmp), _mm_and_si128(mask, sc_N_)); #endif _mm_storeu_si128((__m128i*)((int8_t*)s
+ t), tmp); } } else { for (t = st0; t <= en0; ++t) ((uint8_t*)s)[t] = mat[sf[t] * m + qrr[t]]; }
```

Fixed version that uses `lf_start` and `lf_en`:

```
// loop fission: set scores first int lf_start = st, lf_en = en; if (!(flag &
KSW_EZ_GENERIC_SC)) { for (t = lf_start; t <= lf_en; t += 16) { __m128i sq, st, tmp,
mask; sq = _mm_loadu_si128((__m128i*)&sf[t]); st =
_mm_loadu_si128((__m128i*)&qrr[t]); mask = _mm_or_si128(_mm_cmpeq_epi8(sq,
m1_), _mm_cmpeq_epi8(st, m1_)); tmp = _mm_cmpeq_epi8(sq, st); #ifdef __SSE4_1__
tmp = _mm_blendv_epi8(sc_mis_, sc_mch_, tmp); tmp = _mm_blendv_epi8(tmp,
sc_N_, mask); #else tmp = _mm_or_si128(_mm_andnot_si128(tmp, sc_mis_),
_mm_and_si128(tmp, sc_mch_)); tmp = _mm_or_si128(_mm_andnot_si128(mask,
tmp), _mm_and_si128(mask, sc_N_)); #endif _mm_storeu_si128((__m128i*)((int8_t*)s
+ t), tmp); } } else { for (t = lf_start; t <= lf_en; ++t) ((uint8_t*)s)[t] = mat[sf[t] * m +
qrr[t]]; }
```

Models for additional GPUs

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

pacbio_germline Reference

Run the germline pipeline from FASTQ to VCF by aligning long read sequences with minimap2 and using a deep neural network analysis.

Input/Output file options

--ref REF

Path to the reference file. (default: None)

Option is required.

--index INDEX

Path to a minimizer index file generated by vanilla minimap2 to reduce indexing time. (default: None)

--in-fq IN_FQ

Path to a query sequence file in fastq or fastq.gz format. (default: None)

Option is required.

--knownSites KNOWNSITES

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

--interval-file INTERVAL_FILE

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

--pb-model-file PB_MODEL_FILE

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

--out-recal-file OUT_RECAL_FILE

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

--out-bam OUT_BAM

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

Option is required.

--out-variants OUT_VARIANTS

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

Option is required.

--out-duplicate-metrics OUT_DUPLICATE_METRICS

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

--proposed-variants PROPOSED_VARIANTS

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

Tool Options:

--preset PRESET

Which preset to apply. Possible values are {map-pbmm2,map-hifi,map-ont}. (default: map-pbmm2)

--eqx

Write =/X CIGAR operators. (default: None)

-L INTERVAL, --interval INTERVAL

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

-ip INTERVAL_PADDING, --interval-padding INTERVAL_PADDING

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

--standalone-bqsr

Run standalone BQSR after generating sorted BAM. This option requires both --knownSites and --out-recal-file input parameters. (default: None)

--read-group-sm READ_GROUP_SM

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

--read-group-lb READ_GROUP_LB

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

--read-group-pl READ_GROUP_PL

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

--read-group-id-prefix READ_GROUP_ID_PREFIX

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

--disable-use-window-selector-model

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

--gvcf

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

--norealign-reads

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

--sort-by-haplotypes

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

--keep-duplicates

Keep reads that are duplicate. (default: None)

--vsc-min-count-snps VSC_MIN_COUNT_SNPS

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

--vsc-min-count-indels VSC_MIN_COUNT_INDELS

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

--vsc-min-fraction-snps VSC_MIN_FRACTION_SNPS

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

--vsc-min-fraction-indels VSC_MIN_FRACTION_INDELS

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

--min-mapping-quality MIN_MAPPING_QUALITY

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

--min-base-quality MIN_BASE_QUALITY

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

--alt-aligned-pileup ALT_ALIGNED_PILEUP

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

--variant-caller VARIANT_CALLER

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

--add-hp-channel

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

--parse-sam-aux-fields

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

--use-wes-model

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

--include-med-dp

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

--normalize-reads

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

--pileup-image-width PILEUP_IMAGE_WIDTH

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

--channel-insert-size

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

--no-channel-insert-size

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

--max-read-size-512

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

--prealign-helper-thread

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

--track-ref-reads

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

--phase-reads

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

--dbg-min-base-quality DBG_MIN_BASE_QUALITY

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

--ws-min-windows-distance WS_MIN_WINDOWS_DISTANCE

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

--channel-gc-content

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

--channel-hmer-deletion-quality

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

--channel-hmer-insertion-quality

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

--channel-non-hmer-insertion-quality

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

--skip-bq-channel

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

--aux-fields-to-keep AUX_FIELDS_TO_KEEP

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

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

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

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

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

--consider-strand-bias

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

--p-error P_ERROR

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

--channel-ins-size

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

--max-ins-size MAX_INS_SIZE

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

--disable-group-variants

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

--filter-reads-too-long

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

--haploid-contigs HAPLOID_CONTIGS

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

Performance Options:

--num-threads NUM_THREADS

Number of processing threads. (default: 128)

--gpuwrite

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

--gpuwrite-deflate-algo GPUWRITE_DEFLATE_ALGO

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

--gpusort

Use GPUs to accelerate sorting. (default: None)

--use-gds

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

--low-memory

Use low memory mode (default: None)

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

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

--num-streams-per-gpu NUM_STREAMS_PER_GPU

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

--run-partition

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

--gpu-num-per-partition GPU_NUM_PER_PARTITION

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

--max-reads-per-partition MAX_READS_PER_PARTITION

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

--partition-size PARTITION_SIZE

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

--read-from-tmp-dir

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

Common options:

--logfile LOGFILE

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

--tmp-dir TMP_DIR

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

--with-petagene-dir WITH_PETAGENE_DIR

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

--keep-tmp

Do not delete the directory storing temporary files after completion.

--no-seccomp-override

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

--version

View compatible software versions.

GPU options:

--num-gpus NUM_GPUS

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