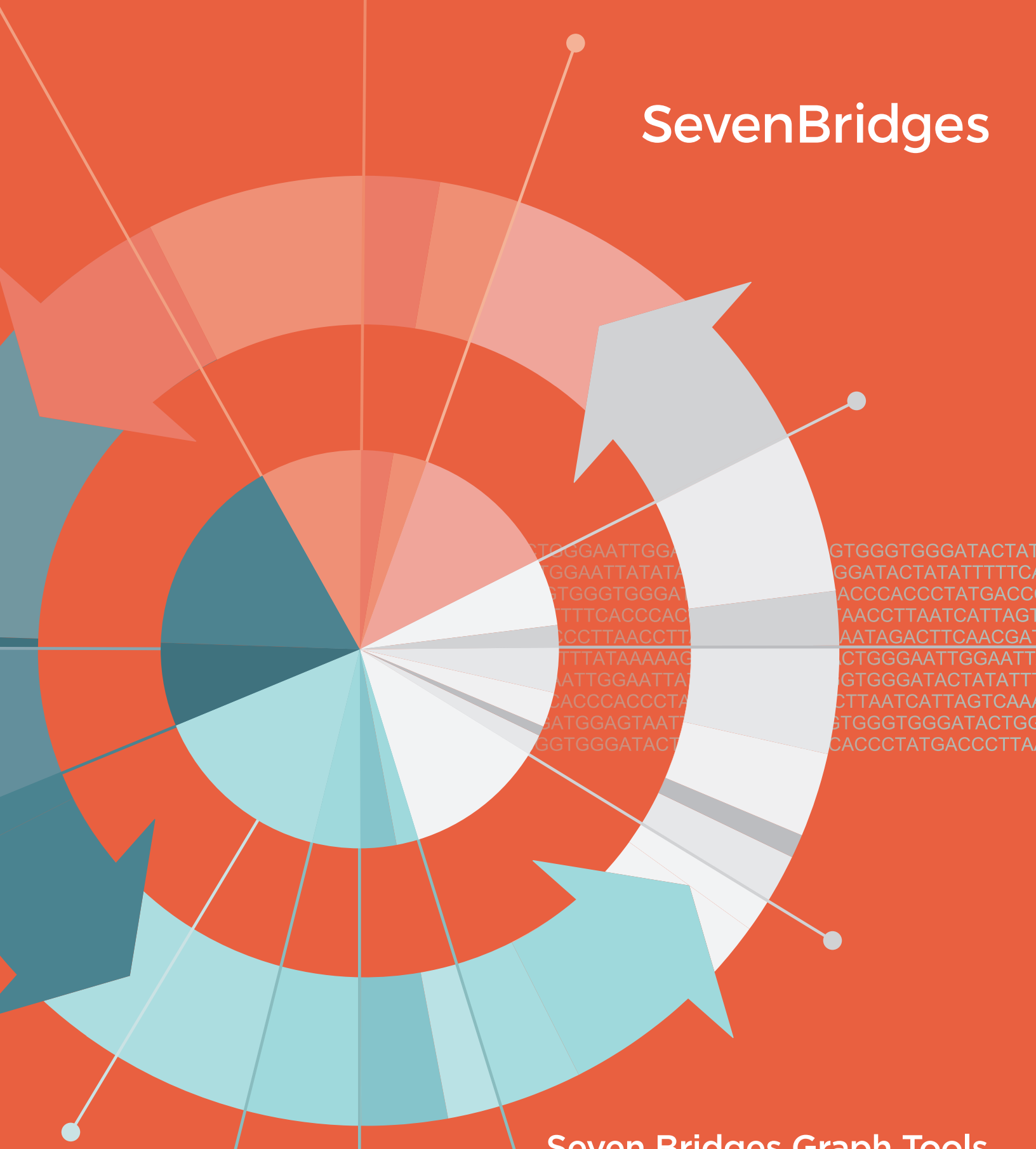
SevenBridges



Seven Bridges Graph Tools
USER GUIDE

V4 (internal numbering)
April, 2020

SevenBridges®

Seven Bridges Graph Tools USER GUIDE

V4 (internal numbering)
April, 2020

VERSION INFORMATION

Version	Date	Description
1	April 12, 2018	First Release
2	March 14, 2019	Second Release
3	December 14, 2019	Added somatic variant calling descriptions
4	April 27, 2020	Updated workflow descriptions

TABLE OF CONTENTS

ACRONYM AND ABBREVIATIONS

SB	Seven Bridges
WGS	Whole Genome Sequencing
WES	Whole Exome Sequence
SNP	Single Nucleotide Polymorphisms
INDEL	Insertion or Deletion Polymorphism
VCF	Variant Call Format
gVCF	Genomics Variant Call Format
BCF	Binary (Variant) Call Format
SAM	Sequence Alignment Map (Format)
BAM	Binary (Sequence) Alignment Map (Format)
BED	Browser Extensible Data

INTRODUCTION

Seven Bridges has developed read alignment and variant calling algorithms that take advantage of sequence and incidence information encoded in Genome Graph References to improve the speed and accuracy of germline and somatic variants inferred from mapping of short reads comprising high-throughput sequencing samples.

This document describes high-performance genome analysis pipelines utilising SB Genome Graph References, as well as the operation of the key components of these pipelines.

GRAPH WORKFLOWS

There are two optimised workflows available for whole genome/exome analysis of germline and somatic (cancer) samples. These workflows are well tested and benchmarked on a variety of real and simulated datasets. Both workflows are available on the Seven Bridges platform and the Cancer Genomics Cloud platform, as well as a CWL implementation for deployment on external compute clusters.

GRAPH GERMLINE PIPELINE

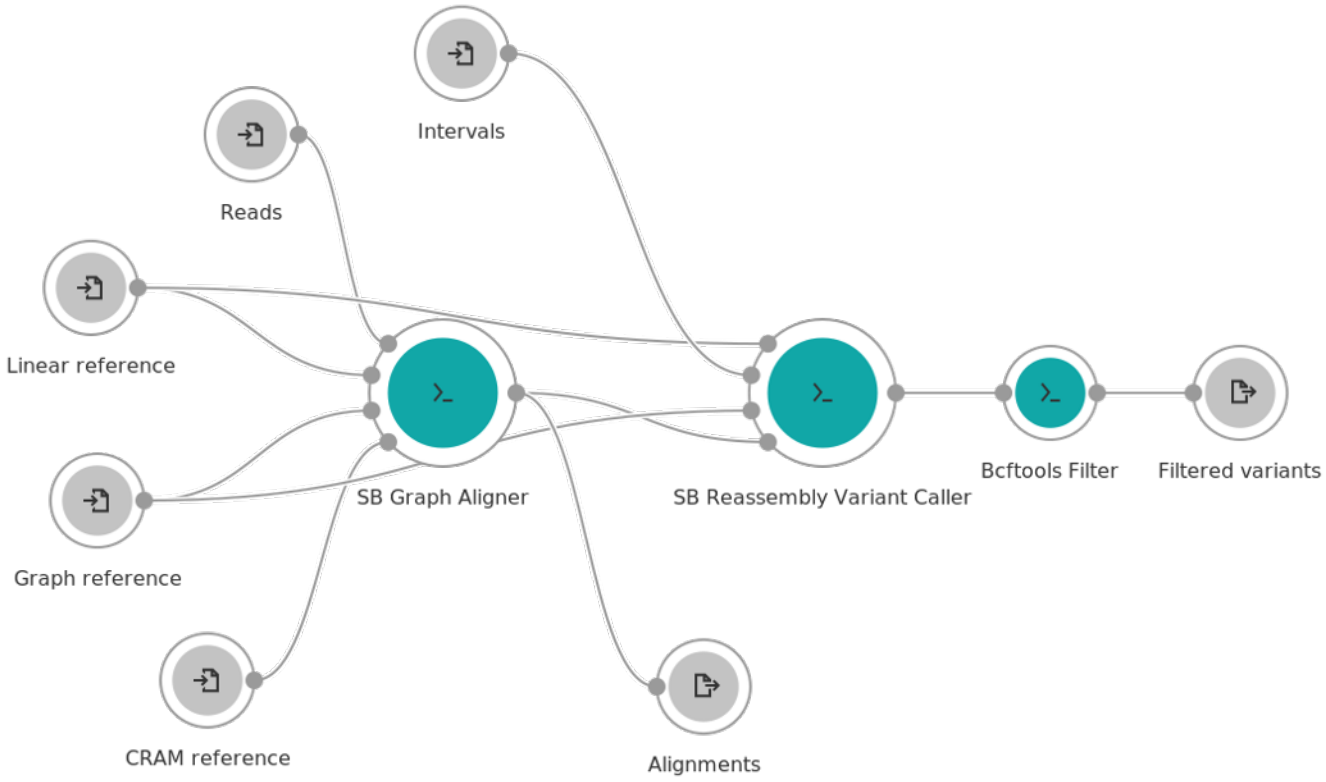


Figure 2. The SB Graph Germline Pipeline

The SB Graph Germline Pipeline is designed to analyze a human sample of short, single-end/paired-end reads either in a distributed cloud environment or a high-performance compute cluster. The SB Graph Germline Pipeline consists of three major steps which are read alignment, germline variant calling and low quality variant filtering (false positive calls).

SB Graph Aligner is a fast and accurate read aligner that maps the input reads on the supplied Genome Graph Reference (specified as FASTA + VCF files) and generates a BAM/CRAM file with graph-related information encoded in custom tags. A more detailed explanation of the tool is given in the SB Graph Aligner (GRAL) section.

SB Reassembly Variant Caller (RASM) finds small variants (SNPs and indels) by utilizing Genome Graph References and reassembling local haplotypes on active regions. A more detailed explanation of the tool is given in the SB Reassembly Variant Caller section.

In the final step of the workflow, Bcftools Filter is run with a pre-defined filter expression to mark likely false-positive artifact calls (as “FP” in the FILTER column).

Workflow specification

Inputs

- **Reads:** The files containing the short DNA sequences for the germline sample. The supported input formats are FASTQ, SAM, BAM and CRAM. If only a single FASTQ file is provided, the reads are treated as single-end. If two FASTQ files are provided, the data is assumed to be paired-end. If the workflow is run on the Seven Bridges platform, the Paired-end metadata field on the FASTQ files must be set as "1" or "2" to denote the pairs of reads prior to task execution. The input files can be gzipped. More than two FASTQ files are not supported. If the given reads are in SAM/BAM/CRAM format (single-end/paired-end information is obtained from 0x1 flag), the reads need to be read name sorted. The following command can be used to sort a SAM/BAM/CRAM file by read name: samtools sort -n. If the supplied input is in CRAM format, the reference file (along with its index) that is used to encode the CRAM file should be given in **CRAM reference** input.
- **Linear Reference:** The linear human reference file is in FASTA format. The graph pipeline supports both GRCh37 (hg19) and GRCh38 (hg38) genome assemblies.
- **Graph Reference:** The graph human reference file in VCF format containing the variants used in genome graph construction. The constructed genome graph is subsequently used in read alignment and variant calling. Both GRCh37 (hg19) and GRCh38 (hg38) are supported.
- **CRAM reference:** The CRAM linear reference file in FASTA format that is used to encode the input reads (when in CRAM format). The FASTA file needs to be indexed. The following command can be used to index a FASTA file: samtools faidx.
- **Intervals:** The target regions for variant calling in BED format. This BED file is also used to parallelize variant calling in a multithreaded environment.

App Settings

No app settings are exposed, the app can be edited on the Seven Bridges platforms or Rabix Composer to expose/change any settings to their non-default values.

Outputs

- **Alignment:** The coordinate-sorted aligned reads in BAM/CRAM format. An accompanying index file in BAI/CRAI format is also outputted. Unaligned reads are placed near their mates if their mates are aligned. If both mates are unaligned, they are added to the end of the BAM/CRAM file. Any duplicates are also marked.
- **Filtered variants:** The final list of variants, in VCF/gVCF format, obtained after variant calling and filtering. Filtered variants are marked as FP in the FILTER column.

GRAPH SOMATIC PIPELINE

Graph Somatic Pipeline is designed to improve somatic calling by constructing a personal genome graph from variants called in the germline sample using global population genome graph, and passing this personal graph to accurately map the tumor and normal samples in order to identify the somatic variants.

Since the personal genome graph includes normal sample's germline variants, it is expected that reference bias is reduced during re-alignment of samples. Re-aligned normal and tumor samples are input to SB Somatic Variant Caller that applies local reassembly to find candidate somatic variants. Personal genomics graph is used to give somatic events a higher prior probability to be a germline variant during genotyping.

After this step, a hard filter is applied to distinguish and eliminate false positive somatic calls. This filter considers variant annotations such as QUAL, MQ, MMQ, MBQ and FS (See Variant Annotator options section for annotation descriptions) to detect low confident somatic calls.

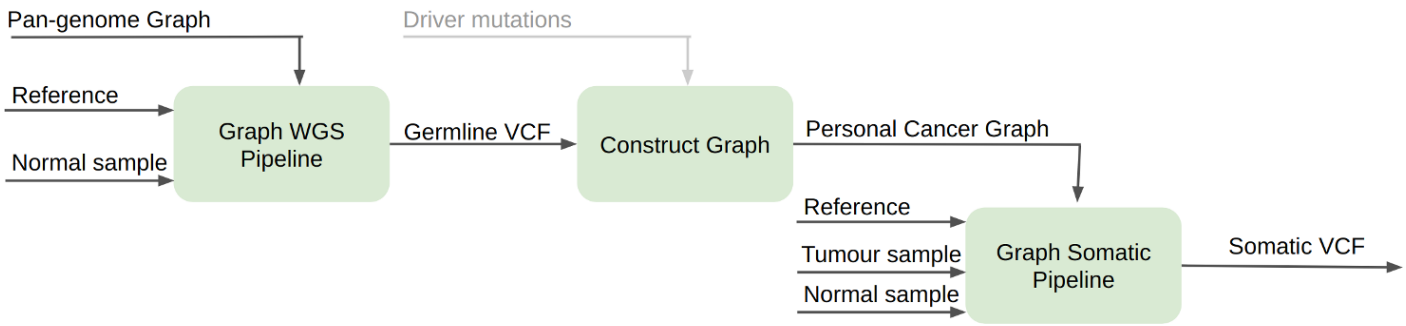


Figure 3: Schematic showing SB Graph Somatic Pipeline

Workflow specification

Inputs

- **Tumor Fastq:** Files containing the short DNA sequences of tumor tissue sample in FASTQ format. Reads should be pair-ended and should be specified in two input ports: Tumor_fastq_1 and Tumor_fastq_2. Multiple files can be used in Tumor_fastq_1 and Tumor_fastq_2 inputs. The input files can be gzipped.
- **Normal Fastq:** Files containing the short DNA sequences of normal tissue sample in FASTQ format. Reads should be pair-ended and specified in two input ports: Normal_fastq_1 and Normal_fastq_2. Multiple files can be used in Normal_fastq_1 and Normal_fastq_2 inputs. The input files can be gzipped.
- **Linear Reference:** The linear human reference file is in FASTA format. Graph pipeline supports both GRCh37 (hg19) and GRCh38 (hg38) genome assembly. The default reference files are public and available here.
- **Graph Reference:** The graph-based reference file. This reference file is specifically designed for use by the SBG graph aligner and the SBG graph variant caller. The latest version of the SBG public graph is available as a default setting.
- **Regions:** The regions in BED format where the variant calling will be performed. This BED file is also used to parallelize the SBG graph variant caller. The recommended input for the whole genome sequencing is available here.

App Settings

Following application settings are optional:

- **Remove duplicates:** Select whether duplicate reads should be removed from the input data. If true, an additional step is run after alignment to remove duplicate reads. This step is recommended if the sequencing library was prepared with polymerase chain reaction (PCR). This step extends the execution time of the pipeline by approximately an hour.
- **Germline call confidence:** Minimum confidence level for a germline variant to be included in the personal graph, phred-scaled threshold.
- **Somatic call confidence:** Minimum confidence level for a somatic variant to be called, phred-scaled threshold.
- **Filter Exclude Expression:** It is possible to define filter conditions by filter exclude expression. Any of the annotations that are described in Variant annotator options section can be used in this expression.
- **Tumor Sample ID:** This sample name should be provided for graph aligner that takes tumor sample as input.
- **Output Names:** It is possible to specify output name of normal and tumor BAM files through these settings. If omitted, output names are generated automatically.

Outputs

- **Tumor Sample Alignment:** The indexed BAM file for tumor sample, aligned against the generated personal graph. Unaligned reads are placed near their mates if their mates are aligned. If both mates are unaligned, they are written to the end of the BAM file.
- **Normal Sample Alignment:** The indexed BAM file for normal sample, realigned against the generated personal graph. Unaligned reads are placed near their mates if their mates are aligned. If both mates are unaligned, they are written to the end of the BAM file.
- **Somatic Variants:** VCF file with results of somatic variant calling. Filtered variants are marked as FP in the FILTER column.
- **Personal Graph Reference:** VCF with germline variants called on the normal tissue sample, used to construct the personal graph for the patient. This personal graph reference is further used for realignment of normal and tumor samples and somatic variant calling in workflow.

GRAPH TOOLS

Seven Bridges Graph Genome Analysis Workflows described above include several custom tools. We described the operation of the key components of the workflows here.

GRAPH ALIGNER (gral)

gral is a fast and accurate read aligner that works on Genome Graphs. Genome Graphs are usually constructed from, but not limited to, a linear reference plus a VCF file containing known variants. Briefly, the alignment algorithm operates on each read (or read pair) as follows:

- A global search is performed to identify candidate regions (seeds)
- A gapless alignment algorithm allowing configurable number of mismatches is used to find the read alignment to the graph for each seed
- If gapless algorithm didn't produce an alignment of sufficient quality, a slower alignment algorithm with support for gaps is applied.
- From alignments found for all seeds, the best one is chosen as follows: For single end reads, the alignment with the best Smith-Waterman score is returned. For paired end reads, the proper pair with the highest total score is returned, unless there is an unpaired alignment that exceeds the paired alignment score by at least unpaired_penalty threshold.
- Graph alignment is projected onto linear reference and written to BAM file

Usage

Input/Output Options

Option	Parameters	Description
-g, --in_graph	<file>	Input graph in binary format
-f, --reference	<file> or <file>.gz	Reference assembly in FASTA format (for graph backbone)
-v, --vcf	<file> or <file>.gz	Variant file in VCF or VCF.GZ format containing graph variants.
-q, --fastq	<file> or <file>.gz	FASTQ file with input reads (containing first read of the pair for paired end reads, unless interleaved format is requested, containing all reads otherwise)
-Q, --fastq2	<file> or <file>.gz	FASTQ file with input reads containing second reads of the pairs
-i, --interleaved_FQ		Treat single input FASTQ as interleaved paired-end reads.
--insert_length	<number>	Skip PE insert length estimation, use provided number for expectation.
--insert_length_sd	<number>	Skip PE insert length estimation, use provided number for standard deviation
-G, --out_graph	<file>	Save reference graph as a binary file
-o, --output	<file>	Output BAM file name. If 'stdout' is specified, redirect output to stdout instead of writing to a file
-n, --skip_reads	<number>	Skip first number reads or read pairs

Option	Parameters	Description
-N	<number>	Stop after aligning number reads or read pairs
--sort		Produce coordinate-sorted BAM file
--keep		Do not delete temporary files after sort completed
--compress	<number>	Set compression level for BAM to number.
--index		Create BAM index. Can only be applied to coordinate-sorted BAM files.
--markdup		Mark duplicates. Can only be applied to coordinate-sorted BAM files.
--read_group_id	<string>	Use string as the read group ID for all reads. If not specified but any read group parameters are provided, the read group ID will be set to random string.
--read_group_sample	<string>	Use string as the read group sample ID
--read_group_unit	<string>	Use string as the read group unit ID
--read_group_library	<string>	Use string as the read group library ID
--reads_group_platform	<string>	Use string as the read group platform. Platform should be as per BAM specification.
-t, --nthreads	<number>	Use number threads for alignment. If omitted, the number of detected CPU logical cores will be used.
--version		Print version info and exit
-h, --help		Print usage information and exit

Notes

- Input graph can be provided either by in_graph option with a binary graph format or by using fasta in combination with vcf. Omitting vcf will result in aligning onto linear reference. in_graph and reference options are mutually exclusive.
- Generated graph can also be outputted in binary format with out_graph option.
- On the platform, read_group_unit, read_group_library, read_group_platform, read_group_id and read_group_sample will be derived from FASTQ file metadata, unless explicitly overwritten. If ony of the parameters is present but read_group_id is omitted, read_group_id will be set to random string. The @RG line will appear in BAM header and RG tag in alignment records only if any of the read group parameters is specified.
- interleaved_FQ will use a single FASTQ file as paired end input (fastq) where reads are interleaved.
- uncompressed will output a BAM with no compression. This is used in addition to output stdout to enable sorting with sambamba as a pipe operation.
- See BAM Tags section for what is controlled by debug_tags.
- insert_length and insert_length_sd are used in both global search windows and compatibility calculations (i.e. deciding if a pair alignment is proper)

Global Search Engine Options (advanced)

Option	Parameters	Description
--hash_table_size_log2	<number>	Index hash table size. If 0, set to an optimal value based upon hash parameters and graph size. Default: 30
--hash_block_step		Interval between successive k-mer blocks to be indexed. Default: 7
--hash_block_size		Size of k-mer blocks to be indexed. Default: 21
--max_match_list_size_deviation	<double>	Hash block occurence for index. Default: 500
--desized_num_number_of_matches	<number>	Estimated hash hits per hash_block_step. Default 4/20 for SE/PE

Notes

- Reference indexing is performed with k-mers of size hash_block_size (default: 21) with intervals of hash_block_step (default: 7).
- Search engine removes hash entries if a particular hash is observed too often in the reference, since common k-mers are not very informative. max_match_list_size_deviation controls the threshold where this happens.
- Search engine will perform search in steps prioritizing unique hits first rather than all possible hit locations. desired_min_number_of_matches defines a threshold where the better (small list) hits are estimated to return a candidate region. If it fails, search engine will fall back to all possible hits for searching candidate regions.
- See this document for a more detailed explanation of these parameters.

Local Alignment Options (advanced)

Option	Parameters	Description
--request_seeds	<number>	Request number seeds from global search. Default: 250
--hash_short_list	<number>	Use only first number seeds for alignment. Default: 50
--min_mismatches	<number>	Allow no more than number mismatches in gapless alignment. Default: 3
--min_glia_score	<number>	Require no less than number Smith-Waterman score in glia alignment. Unqualifying reads will be left unmapped. Default: read_length/3
--glia_score_match	<number>	Smith-Waterman gain for match; deafult: 1
--glia_score_mismatch	<number>	Smith-Waterman penalty for mismatch; default: 4
--glia_score_open_gap	<number>	Smith-Waterman penalty for opening gap; default: 6
--glia_score_extend_gap	<number>	Smith-Waterman penalty for extending gap; default: 1
--unpaired_penalty	<number>	Penalty for not having a proper pair. Default: 17

Other

Option	Parameters	Description
-t, --nthreads	<number>	Use number threads for alignment. If omitted, the number of detected CPU logical cores will be used.
--hts_threads	<number>	Use number CPU threads for BAM compression
--merge_threads	<number>	Use number threads for BAM decompression when merging partially sorted BAM files.
--sort_mem	<number>	Reserve number MB of memory for sorting.
--tmp	<string>	A pattern for creating temporary files during sort operation. Default is '/tmp/sort.chunk.XXXXXX'.
--version		Print version info and exit
--version_json		Print extended version info in JSON format and exit
-h, --help		Print usage information and exit

Option	Parameters	Description
--threads	<INT>	# of running threads
--help		help
--version		Displays version information and exits
--version_json		Output a json representation of the version information

BAM tags

Alignment Tags

The following tags are set for all reads.

- **NM:** Number of mismatches against the reference.
- **YQ:** The Smith-Waterman score of the alignment.
- **Yq:** The suboptimal Smith-Waterman score (i.e. the best Smith-Waterman score for the potential alignments not selected).
- **UQ:** The quality score associated with the seed used in alignment by global search.

Graph Alignment Tags

The following tags are added to each read for graph-related information:

- **XA:** The start locus in the graph relative to the position of the reference alignment on the backbone. This tag is only present if the start locus is not on the linear reference.
- **XB:** The path of the alignment from the start locus. This tag is only present if the alignment takes a non-primary path.
- **XG:** The cigar describing the alignment of the read against the graph path. This tag is only present if the alignment path contains at least one edge that is not on the backbone, or the graph alignment has at least one mismatch.
- **XM:** The number of mismatches against the graph. The tag is only present if that number is different from the mismatches against the reference.

Examples

- Load graph reference from FASTA and VCF:
- aligner --reference reference.fasta --vcf variants.vcf [...]
- Load graph reference from a binary graph file:
- aligner --in_graph graph.gg [...]
- Single-end alignment
- aligner --reference ref.fasta --vcf var.vcf --fastq single-end.fastq --output output.bam [...]
- Paired-end alignment
- aligner --reference ref.fasta --vcf var.vcf --fastq pair-1.fastq --fastq2 pair-2.fastq --output output.bam [...]
- Paired-end interleaved FASTQ
- aligner --reference ref.fasta --vcf var.vcf --fastq pair-interleaved.fastq --interleaved_FQ --output output.bam [...]

Notes on Behavior

Memory

Indexing a whole human genome with a VCF containing ~15 million variants requires about 20GB of memory

Speed

Aligning 50x coverage paired end reads on the whole human genome takes about 7h on a node with 36 cores.

Search Index Parameters

Defaults for search engine are optimized for human genome with 100-150bp long reads. SBG Graph team can advise for parameter values appropriate for sequencing data with different characteristics.

Insert Size Estimation

For paired-end reads, first 10,000 pairs will be used to estimate the insert size mean and standard deviation. These statistics will be used in determining window length for seed lookup, compatibility calculations (properness, i.e. 0x02 flag) and pair penalty. Alternatively, If these values are known beforehand they can be provided with insert_length and insert_length_sd options.

REASSEMBLING VARIANT CALLER (rasm)

rasm is a fast SNP and INDEL caller implementing local reassembly of haplotypes on active regions. It is designed to work with Genome Graph References, usually constructed from a linear reference and a VCF file containing known variants. When using BAM files generated by Seven Bridges Graph Aligner, rasm is able to utilise the embedded graph information to boost sensitivity. Briefly, the variant calling algorithm works as follows:

- Input BAM file is traversed sequentially to identify active regions containing potential variants, detected based on cigar strings of mapped reads.
- An acyclic De Bruijn Graph is constructed from the reads overlapping active region.
- Needleman-Wunsch algorithm is used to find locations of potential variants in each haplotype implied by the traversal from start to end nodes in the De Bruijn graph.
- Events of different haplotypes are combined when they occur at the same locus on the reference.
- A Pairwise Hidden Markov Model is used to calculate scores of haplotype-read pairs for genotype determination.
- Probability that an event is a germline variant is calculated and thresholded based on the scores of the haplotype-read pairs in the corresponding pileup.
- Found variants are written to VCF file with additional annotations.

Usage

Input/Output Options

Option	Description
-?, -h, --help	display usage information
-a, --interval-file	interval file
-b, --bam <FILE>	bam file name
-c, --conservative-mode	turn on conservative mode (no improvements)
-d, --debug	output log files for debugging
-e, --error-model <FILE>	error model file name
-f, --fasta <FILE>	fasta file name
-g, --graph-vcf <FILE>	graph reference in vcf format
-H, --max-haplotypes <INT>	maximum number of haplotypes, default is 20
-i, --interval <INTERVAL>	interval to process, if not specified whole bam file is processed

Option	Description
-j, --combine-unrestricted	allow unrestricted combining, default is off
-k, --max-reassembly-window-size <INT>	maximum variant size
-l, --left-align	left align variants
-m, --no-data-limit	turn off read limits
-n, --per-site-limit <INT>	set per site read limit
-o, --total-read-limit <INT>	set total read limit
-p, --string-overlap-assembler	use string overlap assembler
-q, --gatk-like-genotyping	use GATK HaplotypeCaller-like genotyping
-s, --var-call-threshold <DOUBLE>	variant calling threshold for model 5, phred-scaled default is 30
-r, --reassembly-interval <INTERVAL>	reassembly interval
-t, --threads <INT>	maximum thread count. If the value is 0 thread count will be determined automatically
-v, --vcf <FILE>	vcf file name to output
-w, --wide-format	report ref and alt in wide (untrimmed) format
-x, --annotation-list <ANNOTATIONS>	list of annotations
-y, --mq-threshold <INT>	mapping quality threshold for annotations
-z, --gvcf	gvcf mode
-B, --bamout	generate BAM files that contains haplotypes and realigned reads
-u, --turnoff-genotyping	turn off genotyping of whole haplotypes
-P, --use-af-priors	use graph allele frequency (AF) based prior probs
--pon-vcf <FILE>	panel-of-normals file name for PON annotation
--popaf-vcf <FILE>	population file name for POP_AF annotation

Notes

- Input .bam file should be sorted and indexed
- -j option turns on restricted combining which only combines alleles if they start at the same place. Alleles are combined if they overlap without this combining scheme.
- The interval can be specified in three ways:
  - -i 20 for procession all of 20th chromosome
  - -i 20:10000 starts at location 10000 of chromosome 20 and does the rest of chromosome 20
  - -i 20:10000-20000 does the interval in chromosome 20 of 10000 to 20000

- Multiple intervals can be defined by -i 20:10000-20000 -i 3:300000 -i 21:10000 which processes the interval on chromosome 20 first, then the interval on chromosome 3, and finally the interval on chromosome 21. The interval is inclusive of the first value and exclusive of the last value.
- Total read limit specifies the number of reads to hold in the memory. Total read limit per site is the maximum number of reads to process for any site. If either of these limits is exceeded, extra reads are discarded.
- -l option enables left most selection on multiple variant expression sites
- -H option defines the maximum haplotypes size, if found paths in De Bruijn Graph exceeds this limit, paths are pruned to match the limit.
- -r option enables to perform reassembly in specified interval that can be used for debugging purposes.
- -z option enables gVCF output mode which includes additional information blocks for supporting reference on genome.

Variant Annotator Options

Option	Description
-x	List of annotations; options are default, all and comma separated annotation names
-y <INT>	Minimum mapping quality for reads, default is 20

Notes

- -x option controls the annotations. The values none, all or default can be used or individual annotations can be specified by using the key name. The annotation key names are:
- Default group
  - ‘dp’ for read coverage
  - ‘ad’ for allele depth
  - ‘baseqranksum’ for Wilcoxon rank sum test of base qualities
  - ‘readposranksum’ for Wilcoxon rank sum test of read positions
  - ‘mqranksum’ for Wilcoxon rank sum test of mapping qualities
  - ‘mmq’ for RMS mapping quality
  - ‘qd’ for quality by depth
  - ‘fs’ for Fisher strand
  - ‘sor’ for symmetric odds ratio of strands
  - ‘mnm’ for the mean edit distance for ref and alt supporting reads
  - ‘mme’ for the mean number of mutations for ref and alt supporting reads
- Additional annotations
  - ‘ad_ratio’ for Allele depth ratio
  - ‘hrun’ for homopolymer run counts
  - ‘hrun_max’ for max homopolymer run counts
  - ‘hrun_total’ for total homopolymer run counts

- 'mmq' for mean mapping quality
- 'mbq' for mean base quality
- 'lclip' for number of clips on left side
- 'rclip' for number of clips on right side
- 'clip' for total number of clips
- 'clip_max' for maximum number of clips
- 'csup' for number of reads that variant position is soft-clipped
- 'ru' for tandem repeat unit (bases)
- 'str' for variant is a short tandem repeat
- 'rpa' for the number of times tandem repeat unit is repeated, for each allele (including reference)
- 'sb' for strand bias by sample
- 'mpos' for median distance from end of read
- 'il for median insert length
- 'urc' for the number of unique reads supporting alternate allele
- 'likelihoodranksum' for Wilcoxon rank sum test of alternate and reference haplotype likelihoods
- 'refbases' for the local reference bases
- 'pon' for the site found in panel-of-normals file
- 'popaf' for the frequency of alt alleles (-log10 scale)

Other Options

Option	Description
-h	help

Examples

- Running RASM with required parameters:
  ./somatic-rasm -b path_to_bam_file.bam -f path_to_reference_file.fasta
- Specify output VCF name:
  ./rasm -b path_to_bam_file.bam -f path_to_reference_file.fasta -v out.vcf
- Using Graph Reference:
  ./rasm -b path_to_bam_file.bam -f path_to_reference_file.fasta
  -g path_to_graph_ref.vcf
- Specify interval:
  - Running for only 20th chromosome
    ./rasm -i 20 [...]
  - Running for starting from 10000 of chromosome 20th to end of chromosome
    ./rasm -i 20:10000 [...]

- Running for specific interval such as starting from 10000 to 20000 of chromosome 20

  ./rasm -i 20:10000-20000 [...]

- Running for multiple intervals by binding intervals such as 10000-20000 of 20th chromosome and 10000 to the end of 21th chromosome

  ./rasm -i 20:10000-20000 -i 3:300000 -i 21:10000 [...]

- Add annotations to VCF:
  - Adding default annotations which are DP, AD, SOR, QD, MQ, FS, ReadPosRankSum, MQRankSum, BaseQRankSum

    ./rasm [...] -x default

  - Adding all of annotations (defined in Notes section of Variant Annotator Options)

    ./rasm [...] -x all

  - Adding annotations by name such as SOR, MMQ and HRUN_MAX

    ./rasm [...] -x sor, mmq,hrun_max

  - Using reads that have equal or greater than 30 mapping quality for annotation calculations

    ./rasm [...] -x default -y 30

- Getting gVCF output:

  ./rasm -z [...]

SOMATIC VARIANT CALLER (somatic-rasm)

somatic-rasm is a somatic variant caller that extends the rasm core algorithm to identify somatic variants in cancer tissue sample by comparing to the germline sample from the same patient. It is designed to work with Genome Graph References, usually constructed from a linear reference and a VCF file containing known variants. When using BAM files generated by Seven Bridges Graph Aligner, somatic-RASM is able to utilise the embedded graph information to boost sensitivity. Briefly, the somatic variant calling algorithm works as follows:

- Variants are identified in the normal and tumor BAM file using the reassembly-based variant calling algorithm used by rasm.
- Tumor sample is analysed with lower confidence threshold for variant calls, because of the possibility of contamination.
- Variants that are called in both BAM files are discarded as they are likely to be germline variants.
- The remaining somatic variants are written to VCF file with additional annotations.

Usage

Input/Output Options

Option	Description
-?, -h, --help	display usage information
-a, --interval-file	interval file
-b, --bam <FILE>	bam file name
-c, --conservative-mode	turn on conservative mode (no improvements)
-d, --debug	output log files for debugging
-e, --error-model <FILE>	error model file name
-f, --fasta <FILE>	fasta file name
-g, --graph-vcf <FILE>	graph reference in vcf format
-H, --max-haplotypes <INT>	maximum number of haplotypes, default is 20
-i, --interval <INTERVAL>	interval to process, if not specified whole bam file is processed
-j, --combine-unrestricted	allow unrestricted combining, default is off
-k, --max-reassembly-window-size <INT>	maximum variant size
-l, --left-align	left align variants
-m, --no-data-limit	turn off read limits

Option	Description
-n, --per-site-limit <INT>	set per site read limit
-o, --total-read-limit <INT>	set total read limit
-p, --string-overlap-assembler	use string overlap assembler
-q, --gatk-like-genotyping	use GATK HaplotypeCaller-like genotyping
-s, --var-call-threshold <DOUBLE>	variant calling threshold for model 5, phred-scaled default is 30
-r, --reassembly-interval <INTERVAL>	reassembly interval
-t, --threads <INT>	maximum thread count. If the value is 0 thread count will be determined automatically
-v, --vcf <FILE>	vcf file name to output
-w, --wide-format	report ref and alt in wide (untrimmed) format
-x, --annotation-list <ANNOTATIONS>	list of annotations
-y, --mq-threshold <INT>	mapping quality threshold for annotations
-z, --gvcf	gvcf mode
-B, --bamout	generate BAM files that contains haplotypes and realigned reads
-u, --turnoff-genotyping	turn off genotyping of whole haplotypes
-P, --use-af-priors	use graph allele frequency (AF) based prior probs
--pon-vcf <FILE>	panel-of-normals file name for PON annotation
--popaf-vcf <FILE>	population file name for POP_AF annotation

Notes

somatic-RASM shares most command line options with RASM, with the following changes

- -T, --tumor <FILE> tumor bam file for somatic calling
- -b, --bam <FILE> normal bam file for somatic calling

Variant Annotator Options

Option	Description
-x	List of annotations; options are default, all and comma separated annotation names
-y <INT>	Minimum mapping quality for reads, default is 20

Notes

- -x option controls the annotations. The values none, all or default can be used or individual annotations can be specified by using the key name. The annotation key names are the same as for RASM, with the following new somatic annotations added:
  - ADN: Read depth for each allele in normal sample
  - ADT: Read depth for each allele in tumor sample
  - GERMQ: Phred-scaled probability that at least one of the alt alleles is a germline variant, -10 * log10(P(germline variant))
- Default group
  - 'dp' for read coverage
  - 'ad' for allele depth
  - 'baseqranksum' for Wilcoxon rank sum test of base qualities
  - 'readposranksum' for Wilcoxon rank sum test of read positions
  - 'mqranksum' for Wilcoxon rank sum test of mapping qualities
  - 'mmq' for RMS mapping quality
  - 'qd' for quality by depth
  - 'fs' for Fisher strand
  - 'sor' for symmetric odds ratio of strands
  - 'mnm' for the mean edit distance for ref and alt supporting reads
  - 'mme' for the mean number of mutations for ref and alt supporting reads
- Additional annotations
  - 'ad_ratio' for Allele depth ratio
  - 'hrun' for homopolymer run counts
  - 'hrun_max' for max homopolymer run counts
  - 'hrun_total' for total homopolymer run counts
  - 'mmq' for mean mapping quality
  - 'mbq' for mean base quality
  - 'lclip' for number of clips on left side
  - 'rclip' for number of clips on right side
  - 'clip' for total number of clips
  - 'clip_max' for maximum number of clips
  - 'csup' for number of reads that variant position is soft-clipped
  - 'ru' for tandem repeat unit (bases)
  - 'str' for variant is a short tandem repeat
  - 'rpa' for the number of times tandem repeat unit is repeated, for each allele (including reference)
  - 'sb' for strand bias by sample
  - 'mpos' for median distance from end of read
  - 'il' for median insert length
  - 'urc' for the number of unique reads supporting alternate allele
  - 'likelihoodranksum' for Wilcoxon rank sum test of alternate and reference haplotype likelihoods
  - 'refbases' for the local ref bases
  - 'pon' for the site found in panel-of-normals file
  - popaf' for the negative-log-10 population allele frequencies of alt allele
  - 'adn' for the read depth of each allele in normal sample
  - 'adt' for the read depth of each allele in tumor sample
  - 'germq' for phred-scaled probability that at least one of the alt alleles is a germline variant

Examples

- Running somatic-RASM with required parameters

  ./somatic-rasm -b path_to_normal_bam_file.bam -T path_to_tumor_bam_file.bam -f path_to_reference_file.fasta

- Specify output VCF name:

  ./somatic-rasm -b path_to_normal_bam_file.bam -T path_to_tumor_bam_file.bam -f path_to_reference_file.fasta -v out.vcf

- Using Graph Reference:

  ./somatic-rasm -b path_to_normal_bam_file.bam -T path_to_tumor_bam_file.bam -f path_to_reference_file.fasta -g path_to_graph_ref.vcf

- Specify interval:
  - Running for only 20th chromosome

    ./somatic-rasm -i 20 [...]

  - Running for starting from 10000 of chromosome 20th to end of chromosome

    ./somatic-rasm -i 20:10000 [...]

  - Running for specific interval such as starting from 10000 to 20000 of chromosome 20

    ./somatic-rasm -i 20:10000-20000 [...]

  - Running for multiple intervals by binding intervals such as 10000-20000 of 20th chromosome and 10000 to the end of 21th chromosome

    ./somatic-rasm -i 20:10000-20000 -i 3:300000 -i 21:10000 [...]

- Add annotations to VCF:
  - Adding default annotations which are DP, AD, SOR, QD, MQ, FS, ReadPosRankSum, MQRankSum, BaseQRankSum, MNM, MME

    ./somatic-rasm [...] -x default

  - Adding all of annotations (defined in the Notes section of Variant Annotator Options)

    ./somatic-rasm [...] -x all

  - Adding annotations by name such as SOR, MMQ and HRUN_MAX

    ./somatic-rasm [...] -x sor, mmq,hrun_max

  - Using reads that have equal or greater than 30 mapping quality for annotation calculations

    ./somatic-rasm [...] -x default -y 30

- Specify number of threads:

  ./somatic-rasm [...] -t 2

SevenBridges®

sevenbridges.com