SevenBridges

Seven Bridges GRAF Tools & Workflows USER GUIDE

V1.1, March 2021

TABLE OF CONTENTS

ACRONYM AND ABBREVIATIONS

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

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 genomic variants inferred from mapping of short reads comprising high-throughput sequencing samples.

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

GRAF WORKFLOWS

There are two optimized workflows available for the analysis of high-throughput sequencing data (e.g., whole-genome, whole-exome, panel). These workflows are well tested and benchmarked on a variety of real and simulated datasets. Both workflows are available on all Seven Bridges platforms, and also as Common Workflow Language implementations for deployment on external compute clusters.

GRAF GERMLINE VARIANT DETECTION WORKFLOW

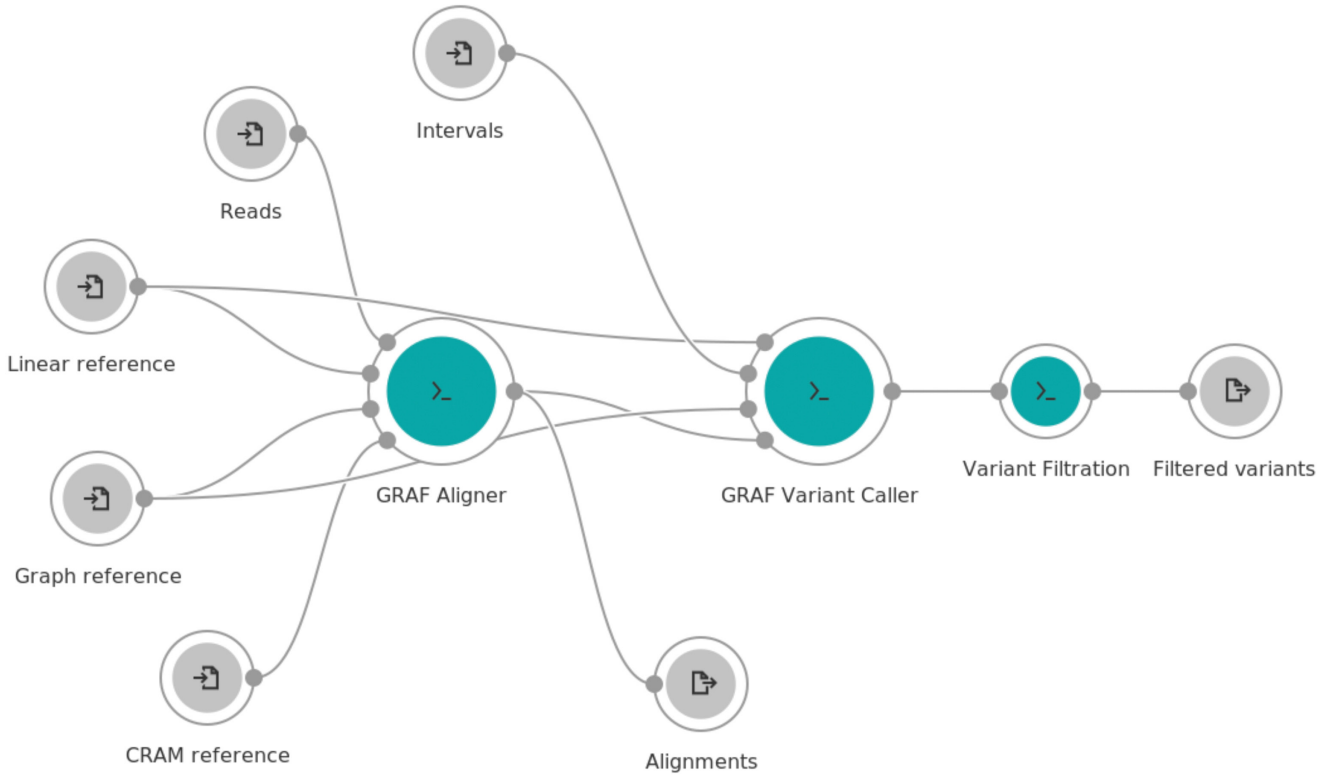

Figure 1. GRAF Germline Variant Detection Workflow

The GRAF Germline Variant Detection Workflow 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 GRAF Germline Variant Detection Workflow consists of three major steps which are read alignment, germline variant calling and low-quality variant filtering (false positive calls).

GRAF 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 GRAF Aligner (GRAL) section.

GRAF 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 GRAF 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. This file must be indexed, with .fai index available in the same path as the FASTA file. The graph pipeline supports both GRCh37 (hg19) and GRCh38 (hg38) genome assemblies.
- **Graph Reference:** The genome graph reference file in VCF format containing the variants used in genome graph construction. This file must be indexed, with .tbi index available in the same path as the VCF file. 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.

GRAF EXTENDED GERMLINE VARIANT DETECTION WORKFLOW

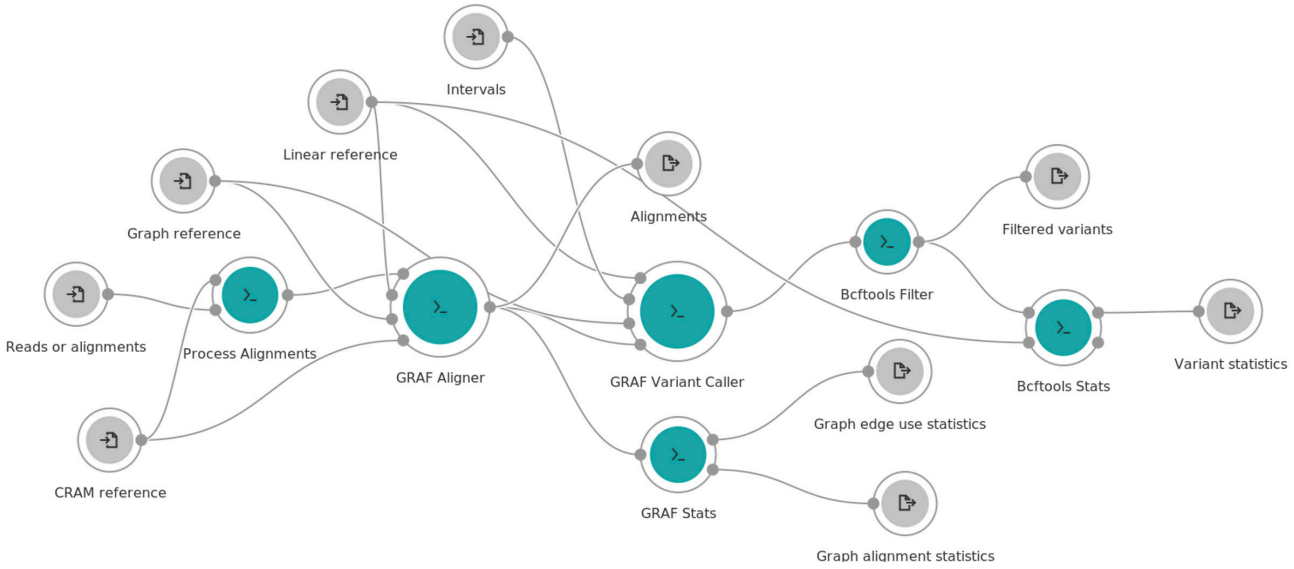

Figure 2: GRAF Extended Germline Variant Detection Workflow

The GRAF Extended Germline Variant Detection Workflow is an extended version of the GRAF Germline Variant Detection Workflow that is designed to directly work with the input sequencing data in FASTQ, SAM, BAM and CRAM formats and produce graph alignment and variant statistics.

The workflow preprocesses the input sequencing data (using Samtools 1.9) to remove secondary and supplementary alignments and sorts the reads by their name which is a common requirement for most aligners. Alignment, variant calling and variant filtration are done using GRAF Aligner, GRAF Variant Caller and Bcftools Filter, respectively, and GRAF Stats and Bcftools Stats are used to generate alignment and variant statistics.

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. Batch task capabilities of the Seven Bridges platform can be used to process multiple samples simultaneously. If the given reads are in SAM/BAM/CRAM format (single-end/paired-end information is obtained from 0x1 flag), the reads will be name-sorted, and the supplementary and secondary alignments will be removed as required by the GRAF Aligner. 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. This file must be indexed, with .fai index available in the same path as the FASTA file. The graph pipeline supports both GRCh37 (hg19) and GRCh38 (hg38) genome assemblies.
- **Graph Reference:** The genome graph reference file in VCF format containing the variants used in genome graph construction. This file must be indexed, with .tbi index available in the same path as the VCF file. 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 Alignment Statistics:** The alignment statistics in JSON format including metrics such as total and aligned number of reads, graph edge use, coverage produced by GRAF Stats tool.
- **Graph Edge Use Statistics:** The detailed statistics of the graph edge use in JSON.GZ format produced by GRAF Stats tool.
- **Variant Statistics:** The variant statistics such as number of SNPs and INDELs, Ti/Tv ratio produced by Bcftools stats.

GRAF MERGE, ANNOTATE, STATS WORKFLOW

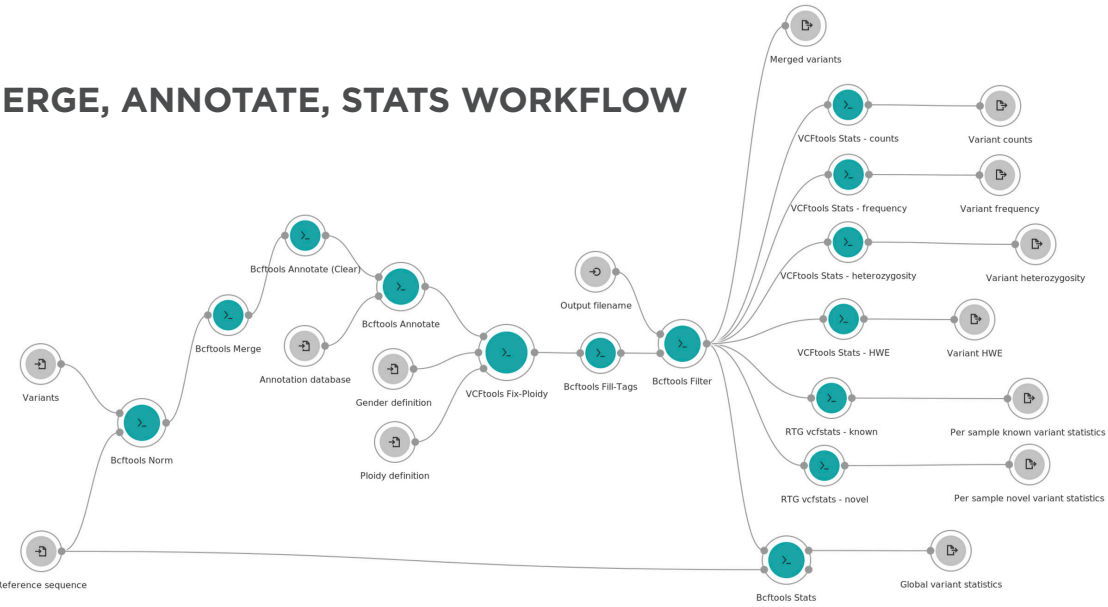

Figure 3: GRAF Merge, Annotate, Stats Workflow

Workflow Specification

The Seven Bridges GRAF Merge, Annotate, Stats Workflow is designed to merge the single sample variants outputted by the GRAF Germline Variant Detection Workflow or the GRAF Extended Germline Variant Detection Workflow, into a multi-sample VCF, annotate the multi-sample VCF file and finally collect various variant statistics on the merged variants.

Inputs

- **Variants:** The list of per-sample VCF files to be merged. All per-sample VCF files must be produced using the same linear reference (FASTA file). The graph reference is not required to be the same.
- **Linear Reference:** The linear reference used to call the sample variants in FASTA format.
- **Gender Definition:** The gender information of the samples given in the variants input. It is a space delimited text file with two columns with sample ID and gender (M/F) per row.
- **Ploidy Definition:** The ploidy information for the reference version (i.e., GRCh38) used in linear reference input. It defines the diploid and haploid regions of the chromosomes including pseudo-autosomal regions (PARs).
- **Annotation Database:** The annotation database (dbSNP) used to annotate variants after merging as known/novel. The variants annotated with a dbSNP identifier are regarded as known variants.

App Settings

- **Output Name:** The filename for the output merged variants.

Outputs

- **Merged Variants:** The merged variants in VCF.GZ format.
- **Global Variant Statistics:** The global variant statistics of the merged variants using Bcftools Stats.
- **Per Sample Novel Variant Statistics:** The per sample novel variant statistics produced by RTG vcfstats.
- **Per Sample Known Variant Statistics:** The per sample known variant statistics produced by RTG vcfstats.
- **Variant Counts:** The number of variants generated by VCFtools.
- **Variant frequency:** The variant frequency information generated by VCFtools.
- **Variant heterozygosity:** The variant heterozygosity information generated by VCFtools.
- **Variant HWE:** The variant Hardy-Weinberg equilibrium (HWE) information generated by VCFtools.

GRAF TOOLS

Seven Bridges GRAF Workflows described above contain several proprietary tools. We describe the operation of the key components of the workflows here.

GRAF 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 the 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
-f, --reference	<file> or <file>.gz	Reference assembly in FASTA format (basis for the graph variants).
-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). The input file may also be in a name-sorted SAM, BAM or CRAM format.
-Q, --fastq2	<file> or <file>.gz	FASTQ file with input reads containing second reads of the pairs. Not used when the input format is SAM, BAM, CRAM or interleaved FASTQ.
-i, --interleaved_FQ		Treat single input FASTQ as interleaved paired-end reads.
--cram_ref	<file> or <file>.gz	Reference assembly that was used to generate the input CRAM.
--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 processing *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 or CRAM index. Can only be applied when output is coordinate-sorted.
--markdup		Mark duplicates. Can only be applied when output is coordinate-sorted.
--trim		Enable adapter sequence detection and trimming. Recommended with exome PE input, unless external trimming algorithm has been applied.
--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.
--version		Print version number and exit.
--version_json		Print extended version and build info in JSON format and exit.
-h, --help		Print usage information and exit.

Notes

- Input graph is provided as Seven Bridges approved (i.e., signed) VCF, FASTA pair.
- 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 any of the parameters is present but read_group_id is omitted, read_group_id will be set to a 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 a paired end input (fastq) where reads are interleaved.
- 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 occurrence 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. Default: 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'.

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 [...]

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

Insert Size Estimation

For paired-end reads, first 50,000 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.

GRAF 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 an 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 a 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)
-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 per site read limit
-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, 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 based on hardware
-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 contain 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
-R, --disable-phasing	Disable default physical phasing in gVCF mode
--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.
- -P option enables to use allele frequency of variants from graph reference (determined by -g option).
- -R option disables the physical (haplotype) phasing, otherwise it is turned on by default in gVCF mode.
- --pon-vcf can be used to define indexed VCF for PON annotation. This annotation only specifies calls that are found in the given pon-vcf by "PON" keyword in the INFO field of relevant calls. Users can apply blacklist variants as PON VCF to annotate found variants.
- --popaf-vcf can be used to define indexed VCF for POPAF annotation. This annotation only specifies negative log10 scale population allele frequencies of calls' alternative alleles which are found in the given population (popaf) VCF.

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
  - 'mq' 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 ($-\log_{10}$ scale)

Other Options

Option	Description
-h	Help

Examples

- Running RASM with required parameters:

  ./rasm -b path_to_bam_file.bam -f reference.fasta -g graph_ref.vcf -v out.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 21st 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 [...]

GRAF STATS

GRAF Stats is designed to generate alignment statistics for the alignments done using linear references as well as graph references.

Usage

Input/Output Options

Option	Description
--bam	Alignments in BAM format (required)
--vcf	Graph reference in VCF format (optional)
--bed	Intervals to generate statistics on (optional)
--output	Output filename prefix (optional, defaults to alignment filename)
--pretty	Indented JSON output (optional, defaults to False)

Notes

Alignment statistics generated, output contains:

- **alignment:** alignment statistics
  - **total:** total number of reads
  - **mapped:** number of mapped reads
  - **unmapped:** number of unmapped reads
- **aligned_length:** query aligned length statistics, i.e., query_length - soft_clips
  - **linear:** with respect to linear reference
    - **mean:** mean of aligned length
    - **std:** standard deviation of aligned length
    - **dist:** aligned length distribution (as a list of (length, count) pairs)
  - **graph:** same division as above
- **flags:**
  - **secondary:** secondary flag is set
  - **qc_fail:** qc_fail flag is set
  - **duplicate:** duplicate flag is set
  - **supplementary:** supplementary flag is set
- **edges:** graph edge statistics
  - **total:** total number of non-backbone edges
  - **used:** number of non-backbone edges used by reads
- **edges_breakdown:** graph edge statistics split into variant types
  - **total:** statistics for all non-backbone edges in the graph. size in distributions is abs(len(ref) - len(alt)) except for SNPs and MNPs where size is the number of bases (i.e., 1 for all SNPs)

        - **SNP:** statistics for SNPs
            - **count:** total number
            - **dist:** size distribution (as a list of (size, count) pairs)
        - **MNP:** statistics for MNPs
            - same division as above
        - **INS:** statistics for simple insertions
            - same division as above
        - **DEL:** statistics for simple deletions
            - same division as above
        - **CINS:** statistics for complex variants where inserted sequence is longer
            - same division as above
        - **CDEL:** statistics for complex variants where inserted sequence is shorter
            - same division as above
    - **used:** statistics for used non-backbone edges in the graph
        - same division as above
- **mapped_bases:** total number of bases mapped against reference in all reads
    - **linear:** with respect to linear reference
    - **graph:** with respect to graph reference; if alignment was done without a graph reference, then this is the same as linear
- **mismatch:** total number of mismatches against reference in all reads
    - **linear:** with respect to linear reference
    - **graph:** with respect to graph reference; if alignment was done without a graph reference, then this is the same as linear
- **error_rate:** fraction of different bases against reference in all reads (mismatch/mapped_bases)
    - **linear:** with respect to linear reference
    - **graph:** with respect to graph reference; if alignment was done without a graph reference, then this is the same as linear
- **reads:** read based statistics
    - **proper:** number of proper reads
        - **total:** total number of reads
        - **backbone:** number of reads aligned against backbone only
        - **non-backbone:** number of reads using a non-backbone edge
    - **improper:** number of improper reads
        - same division as above
- **MQ:** mapping quality statistics
    - **0:** number of reads with mapping quality zero (mq == 0)
        - **total:** total number of reads
        - **backbone:** number of reads aligned against backbone only
        - **non-backbone:** number of reads using a non-backbone edge

    - **20-:** number of reads with mapping quality below 20 (0 <= mq < 20)
        - same division as above
    - **20+:** number of reads with mapping quality equal or above 20 (mq >= 20)
        - same division as above
- **coverage:** coverage statistics
    - **main:** coverage for main chromosomes (1-Y+MT) or BED regions if --bed is given
        - **total_bases:** total number of bases for contigs in the genome
        - **covered:** number of bases covered by reads
        - **average:** average coverage of covered bases
        - **breadth:** fraction of linear reference covered (covered/total_bases)
        - **dist:** coverage distribution (as a list of (coverage, count) pairs)
    - **other:** coverage for other contigs
        - same division as above

Secondary reads are excluded from all statistics except flag counts. supplementary reads are excluded from read counts but are included in other metrics.

Edge statistics generated, each entry contains:

- **variant:** variant information (chrom pos ref alt)
- **lref:** reference span (will be 0 for pure insertions)
- **lalt:** alt sequence length (will be 0 for pure deletions)
- **reads:** number of reads that use the edge
- **coverages:** list of coverages for each base on the edge.

lref and lalt is not necessarily len(ref) or len(alt). While constructing the graph, the same prefix or suffix will be stripped. For simple deletions (or insertions), lalt (or lref) will be 0. In the same manner, coverages list will contain a number of entries equal to lalt. For deletions, coverages will contain a single entry and it will denote the number of reads that use corresponding deletion edge.

SevenBridges®

sevenbridges.com