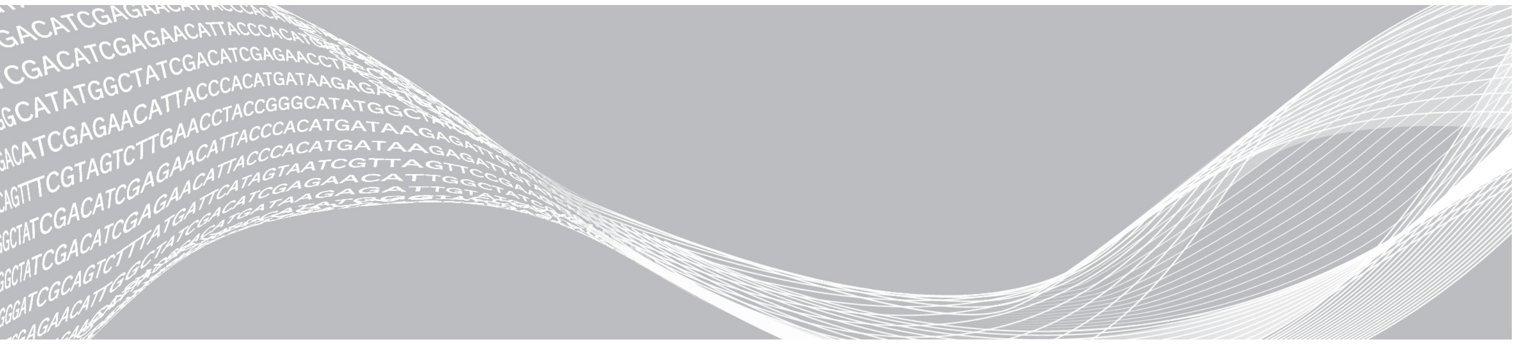
TruGenome Clinical Sequencing Services

User Guide



Document # SAN EXTERNAL-0037 rev A
June 2017

Table of Contents

Chapter 1 Getting Started

Introduction

The TruGenome Clinical Sequencing Service leverages a suite of proven algorithms to detect genomic variants comprehensively and accurately. High-quality sequence reads are aligned using the Isaac Sequence Aligner and Variant Caller. For more information, see *Isaac Aligner* on page 11. Variant calling is performed using the Isaac Variant Caller. For more information, see *Strelka (Small Variant Caller)* on page 12.

Identified variants are then annotated and a summary PDF of the technical data is provided.

The sections that follow provide an overview of the source and contents of the main files that Illumina Clinical Services Lab (ICSL) creates using the informatics pipeline, as well as information about key algorithms, such as the Isaac Variant Caller. This information will help you understand the TruGenome Clinical Sequencing Service data package that you receive from ICSL.

ICSL Hard Drive Data Delivery

ICSL provides data for sequenced and assembled genomes on one or more hard drives. The hard drives are formatted with the NTFS file system and are encrypted using the open-source cross-platform TrueCrypt software and the Advanced Encryption Standard (AES) algorithm (Federal Information Processing Standards Publication 197).

The data on the hard drive are organized in a folder structure with one top-level folder that is named with the barcode of the associated sample. For more information on the folder structure, see the sections that follow.

Analysis Deliverables

Overview

This section details the files and folder structure for the TruGenome Clinical Sequencing Service results. The files and folders are named based on the unique sample identifiers. Usually, these unique identifiers are the barcodes associated with the samples in the lab, but can be a known sample ID for reference samples.

Results Folder Structure

Under each sample folder, you can find the following file structure that contains analysis results.

- [Sample_Barcode]
  - Assembly
    - **[Sample_Barcode].bam**—Archival *.bam file for sample
    - **[Sample_Barcode].bam.bai**—Index for *.bam file
  - Genotyping
    - **[Sample_Barcode].Genotyping.vcf.gz**—Genotyping SNPs mapped to reference in *.vcf format
    - **[Sample_Barcode].GenotypingReport.txt.gz**—Genotyping SNPs tab delimited report
  - Variations
    - **[Sample_Barcode].genome.vcf.gz**—Genome VCF file containing SNPs, indels, and reference covered regions
    - **[Sample_Barcode].json.gz**—JSON file containing all the variants listed in the SampleBarcode.genome.vcf.gz with all associated variant annotation information.
    - **[Sample_Barcode].vcf.gz**—Single nucleotide polymorphisms (SNPs) and small Insertion/Deletion calls in VCF format
- md5sum.txt—Checksum file for confirming file consistency
- [Sample_Barcode].TechnicalReport.pdf—PDF report detailing sample information and statistics

NOTE

All the VCF files that ICSL provides are compressed and indexed using tabix. For details about tabix, see the tabix manual in SAMtools (at http://www.htslib.org/download/).

The tabix index shows up as an additional [Sample_Barcode].TYPE.vcf.gz.tbi file. It can be used for fast retrieval of targeted regions in the associated *.vcf.gz file

Assembly

[Sample_Barcode].bam

The included archival BAM file contains all pass filter reads input into the analysis pipeline for a sample and includes aligned, duplicate, and unaligned reads.

[Sample_Barcode].bam.bai

This file is the index for the BAM file and can be used with SAMtools and other tools utilizing the SAMtools specification for fast retrieval of targeted regions in the associated BAM file.

BAM File Details

The included BAM file adheres to the SAM format specification wherever possible. The following sections cover BAM file details that are not evident in the specification:

- Singleton / Shadow Pairs
- Read Groups: RG
- Read name: RNAME
- Bitwise Flag Notes: FLAG
- Extended Tags / Optional Fields
- MAPQ

Singleton / Shadow Pairs

Singleton/shadow pairs refer to pairs for which the aligner was unable to determine the alignment of one of the ends. The determined end is the singleton and the undetermined end is the shadow. Shadows are assigned the position of the end that does align. To maintain SAMtools format compatibility, the shadows are stored in the BAM file immediately after their respective singletons, with CIGAR empty and corresponding flag (4) set. Shadows can be retrieved using the following SAMtools command:

```
samtools view -f 4 input.bam > output.sam
```

Read Groups: RG

Where possible, unique flow cell-lane-index mappings split up the read groups in the BAM. The following is an example from a BAM header:

```
@RG ID:0 PL:ILLUMINA SM:NA12878 PU:C0L54ACXX_0:1:none
@RG ID:1 PL:ILLUMINA SM:NA12878 PU:C0L54ACXX_0:2:none
@RG ID:2 PL:ILLUMINA SM:NA12878 PU:C0L54ACXX_0:3:none
```

In the example, the read group 0 is derived from the flow cell barcode ID C0L54ACXX, lane 1, without a specified index for sample NA12878. Read groups 1 and 2 are from the same flow cell, lanes 2 and 3. The _0 denotation on the end of the flow cell is reserved for rare cases in which the same flow cell could appear multiple times in the same analysis and is not utilized for the TruGenome Clinical Sequencing workflow.

Read Name: RNAME

The read name consists of the following pattern, which details the flow cell, lane, and tile on which the sample was run:

```
flowcell-id ":" lane-number ":" tile-number ":" cluster-id ":0"
```

ID	Description
flowcell-id	Flow cell barcode.
flowcell-idx	Unique 0-based index of the flow cell within the analysis.
lane-number	Lane number 1–8.
tile-number	Unpadded tile number.
cluster-id	Unpadded 0-based cluster ID in the order in which the clusters appear within the tile.

Bitwise Flag Notes: FLAG

The bitwise flags used are described in the following table.

Bit	Description	Note
0x1	Template having multiple segments in sequencing.	Always set on for paired reads.
0x2	Each segment properly aligned according to the aligner.	Pair matches dominant template orientation.
0x4	Segment unmapped.	Set for unmapped reads.
0x8	Next segment in the template unmapped.	Paired read is unmapped.
0x10	SEQ being reverse complemented.	Read mapped to strand of reference.
0x20	SEQ of the next segment in the template being reversed.	Paired read mapped to strand of reference.
0x40	The first segment in the template.	Read 1 sequence.
0x80	The last segment in the template.	Read 2 sequence.
0x100	Secondary alignment.	Secondary alignments can be found in the BAM file with the tag SA.
0x200	Not passing quality controls.	Nonpass filter reads are not included (always off).
0x400	PCR or optical duplicate.	Read 1 and Read 2 were marked as duplicate reads.

Extended Tags and Optional Fields

The aligner produces the following fields in the BAM file.

Tag	Isaac Definition
AS	Pair alignment score.
BC	Barcode string.
NM	Edit distance (mismatches and gaps) including the soft-clipped parts of the read.
OC	Original CIGAR for the realigned reads. See realign-gaps.
RG	Isaac read groups correspond to unique flow cell lane barcodes.
SM	Single read alignment score.
ZX	Cluster X pixel coordinate on the tile times 100.
ZY	Cluster Y pixel coordinate on the tile times 100.

Mapping Quality (MAPQ)

For pairs that match the dominant template orientation, the MAPQ value in the AS field is capped at 60. For reads that are not members of a pair matching the dominant template orientation, the MAPQ value in the SM field is capped at 60. The MAPQ could be downgraded to 0 or set to be unknown (255) for alignments that do not have enough evidence to be correctly scored.

Genotyping

Samples are genotyped on the Infinium platform. The sequencing SNP calls are checked against the Infinium SNP calls to confirm identity and ensure that the data is of high quality. This folder contains the results of the genotyping SNP calls.

[Sample_Barcode]Genotyping.vcf.gz

This file contains the genotyping SNPs in VCF format. The genotyping SNPs were mapped to the reference using megaBLAST and filtered for unique mappings. The VCF file contains the following fields.

INFO Fields

Field	Description
AL	Array alleles relative to the design strand of the array probe.
ST	The strand for the array alleles relative to the reference. A dash (-) denotes a reverse compliment.

FORMAT Fields

Field	Description
GC	The GenCall score from the genotyping SNP call. (0.15 cutoff applied by default).
GT	Genotype per VCF specification.

FILTER Fields

Field	Description
GTEX	The exclude genotype filter. The genotype was excluded in the mapping, possibly because the probe failed to find a reference map, failed to map uniquely, or was an intensity-only based probe.

Genotyping Report

The [Sample_Barcode].GenotypingReport.txt.gz file contains the genotyping report that is output from the GenomeStudio Genotyping Module. The genotyping report is a tab-delimited text file and includes a header followed by at least the following columns.

Column	Description
Allele1—Design	The A allele call that is relative to the probe.
Allele1—Forward	The A allele call that is relative to the submitted sequence.
Allele2—Design	The B allele call that is relative to the probe.
Allele2—Forward	The B allele call that is relative to the submitted sequence.

Column	Description
GC Score	The GenCall score. This score is a quality metric assigned to every genotype called, and generally indicates its reliability. GC scores have a maximum of 1, and are calculated using information from the clustering of the samples. Each SNP is evaluated based on the angle of the clusters, dispersion of the clusters, overlap between clusters, and intensity. Genotypes with lower GC scores are located furthest from the center of a cluster and have a lower reliability.
Sample Barcode	The internal process identifier.
SNP Name	The SNP identifier. An rsID for dbSNP content.

Variations

The variations folder contains the variant call output in VCF 4.1 format for the sample. Each variant file is compressed and includes an index that was created with tabix, for fast, range-based access. This is a summary of the outputs for each sample. For more information, see *Overview* on page 11

The VCF files are annotated with the Illumina annotation pipeline and contain additional INFO fields pertaining to the annotations. For more information, see *Illumina Annotation Pipeline* on page 18.

[Sample_Barcode].genome.vcf.gz

The genome VCF file contains VCF formatted output for the SNPs, indels, and block compressed nonvariant position output. You can use this file to quickly compare variants and covered regions between samples. The filters and INFO fields are a combination of both the SNP and indel VCF filters listed in the following tables, along with the block compressed specific flags. For more information, see *gVCF (Genome VCF)* on page 14. For additional INFO fields pertaining to annotation information, see *Illumina Annotation Pipeline* on page 18 on page 11

[Sample_Barcode].vcf.gz

This file lists the single nucleotide polymorphisms and indels that were called by the Isaac Variant Caller.

The VCF file contains the following fields.

Info Fields

Field	Description
AF1000G	The allele frequency from all populations of 1000 genomes data.
AA	The inferred allele ancestral (if determined) to the chimpanzee/human lineage.
CIGAR	The CIGAR alignment for each alternate indel allele.
COSMIC	The numeric identifier for the variant in the Catalogue of Somatic Mutations in Cancer (COSMIC) database. Format: GenotypeIndex|Significance.
CSQR	Predicted regulatory consequence type. Format: GenotypeIndex|RegulatoryID|Consequence.
CSQT	Consequence type as predicted by IAE. Format: GenotypeIndex|HGNC|Transcript ID|Consequence.
DUP	Normalized version of this variant has a duplicate.
END	The end position of the region described in this record.
EVS	Allele frequency, coverage and sample count taken from the Exome Variant Server (EVS). Format: AlleleFreqEVS|EVSCoverage|EVSSamples.

Field	Description
GMAF	Global minor allele frequency (GMAF); technically, the frequency of the second most frequent allele. Format: GlobalMinorAllele|AlleleFreqGlobalMinor.
IDREP	Number of times RU is repeated in an indel allele.
MQ	RMS of mapping quality.
NF	Normalization failed for this variant (could not be left-shifted due to conflicting upstream variant).
OLD_ID	The ID field prior to decomposition/normalization.
OLD_VARIANT	Original chr:pos:ref:alt encoding.
phyloP	PhyloP conservation score. Denotes how conserved the reference sequence is between species throughout evolution.
RefMinor	Denotes positions where the reference base is a minor allele and is annotated as though it were a variant.
REFREP	Number of times RU is repeated in the reference.
RU	The smallest repeating sequence unit extended or contracted in the indel allele relative to the reference. If RUs are longer than 20 bases, they are not reported.
SNVHPOL	SNV contextual homopolymer length.
SNVSB	SNV site strand bias.
Unphased	Indicates a record that is within the specified phasing window of another variant, but could not be phased because of a lack of minimum read support.

Format Fields

ID	Description
AD	Allelic depths for the ref and alt alleles in the order listed. For indels, this value includes only reads that confidently support each allele. Specifically, includes reads for which the posterior probability is 0.999 or higher that the read contains an indicated allele versus all other intersecting indel alleles.
ADF	Allelic depths on the forward strand
ADR	Allelic depths on the reverse strand
DP	Filtered base call depth used for site genotyping.
DPF	Base calls filtered from input before site genotyping.
DPI	Read depth associated with indel, taken from the site preceding the indel.
FT	Sample filter. PASS indicates that all filters have passed for this sample.
GQ	Genotype quality.
GQX	Empirically calibrated variant quality score for variant sites, otherwise the minimum of {Genotype quality assuming variant position, Genotype quality assuming nonvariant position}.
GT	Genotype.
PL	Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification.
PS	Phase set identifier.
SB	Sample site strand bias.
VF	Variant frequency.

Filter Fields

ID	Description
ADFilter	Call quality inconsistent with AD.
DPFilter	Call quality inconsistent with DP.
HighDepth	The locus depth is greater than 3× the mean chromosome depth.
HighDPFRatio	The fraction of base calls filtered out at a site is > 0.4.
HighSNVSB	SNV strand bias value (SNVSB) exceeds 10.
IndelConflict	The locus is in a region with conflicting indel calls.
IndelSizeFilter	Indel is outside reportable size range. Insertion/Deletion range reported in VCF header.
LowGQX	Locus GQX is < 30 or not present.
Pass	All filters passed.
PloidyConflict	Genotype call from the variant caller is not consistent with chromosome ploidy.
SiteConflict	The site genotype conflicts with the proximal indel call, which is typically a heterozygous SNV call made inside a heterozygous deletion.

Sample Barcode Technical Report

[Sample_Barcode]TechnicalReport.pdf

This report contains an overview of the results for the samples. The report contains the following information:

- Sample Information
- Library Specifications
- Data Volume
- Passing Filter and Aligned Basecall Quality Score Distribution
- Coverage Summary
- Non-N Reference Coverage Distribution
- SNP/Indel Assessment
- Variant Statistics

Sample Information

This section of the Sample Barcode Technical Report contains information associated with the sample from the included sample requisition. If the sample was de-identified for processing, this section contains minimal content, such as the de-identified sample name and gender.

SNP and Indel Assessment

These two tables in the Sample Barcode Technical Report provide the total number of SNPs and indels overlapping known variants and genes, exons, and coding regions. All counts only use PASS filter variants where applicable.

Value	Description
% in Coding	Percent of PASS filter variants overlapping a coding position for any annotated transcript.
% in dbSNP	Percent of PASS filter variants that overlap a dbSNP identifier in annotation.
% in Exons	Percent of PASS filter variants overlapping a coding, 5' UTR, or 3' UTR position for any annotated transcript.
% in Genes	Percent of PASS filter variants overlapping a coding, 5' UTR, 3' UTR, or intron position for any annotated transcript.
Het/Hom	The ratio of heterozygous to homozygous PASS filter variants reported.
Ti/Tv	The transition/tranversion ratio for reported variants relative to the reference base or bases.
Total	Total number of PASS filter SNVs reported.

Variant Statistics

This table provides a breakdown of SNPs and indels into total counts as well as their annotated consequences. Complex indels are split into deletions and insertions where appropriate. Consequence types for overlapping transcripts are counted under the most severe transcript consequence according to the annotation.

Library Specifications

This section of the Sample Barcode Technical Report describes the details related to the library prep used in the sample.

Value	Description
Fragment Length Median	Median fragment length of library sequence fragments calculated for each pair of mapped reads. For normal reads, this value includes both reads, along with the unsequenced insert between the reads.
Fragment Length SD	The standard deviation of fragment lengths around the median.
Read Length	Read lengths used in the build.
Read Type	Used as paired end for the standard Whole Genome Sequencing workflow.

Data Volume

This table in the Sample Barcode Technical Report reports the volume of data input into the alignment process and in the associated BAM file.

Value	Description
Passing Filter	Yield is the number of gigabases of data (PASS filter data only) input into the build. % Bases ≥ Q30—Q30 is the percent of the sequence data that has a Q-score of Q30 or greater.
Passing Filter and Aligned	Same as passing filter but reported only for the subset of data that aligns to the genome.

Coverage Summary

Coverage summary reports the distribution of depth of coverage across the genome. Coverage is calculated from bases that are not flagged as duplicates, and for which both read pairs map unambiguously.

Value	Description
% ≥ 5/10/20x coverage	Number of non-N reference autosomal positions that have ≥ 5/10/20 fold coverage.
% Callable	The percent of autosomal non-N reference genome in gVCF file with a PASS filter status.
Average Coverage	Mean coverage across the genome defined as "bases uniquely aligned over non-N autosomal regions" / "total non-N reference length of autosomal regions".

Non-N Reference Coverage Distribution

This histogram of coverage depth uses the same definition of coverage as the Coverage Summary.

Data Integrity

The md5sum.txt file is provided to check the consistency of the sample files and folders. Immediately after sample quality check, the md5sums for every file in the directory tree are generated. If media failures compromise data integrity, you can use the md5sum tool to find the inconsistencies. Use the tool to compare the hash from the provided md5sum file to the hash generated from the downloaded file.

On a Unix system, you can use the following commands to perform an md5sum check, assuming the utility is installed:

```
% cd [Sample_Barcode]
% md5sum -c md5sum.txt
```

The check verifies every file in ~30–45 minutes. Any errors are listed in the output.

In Windows, there are various command line and GUI tools available to perform an md5sum check. The Cygwin tools provide a utility identical to Linux.

Analysis Overview

Overview

After the sequencer generates base calls and quality scores during primary analysis, the resulting data are analyzed in two steps; alignment to the reference genome, followed by alignment and variant calling.

Alignment and variant calling are performed with the Isaac Aligner and the Isaac Variant Caller. They produce the following output.

- Realigned and duplicate marked reads in a BAM file format.
- Variants in a VCF file format.
- An additional Genome VCF (gVCF) file. This file features an entry for every base in the reference, which differentiates reference calls and no calls, and a summary of quality.

Genome Specific Details

ICSL uses Ensembl as a reference genome. The chromosome naming scheme follows the Ensembl conventions of chromosomes 1-22, X, Y, MT. The pseudoautosomal region (PAR) of the Y chromosome is masked out with Ns. As a result, any mappings occurring in the PAR region map to the X chromosome. Only the main chromosomes and mitochondria are used in the reference; alternative loci are not included. As per GATK specification for UCSC, chromosome MT is the first chromosome, followed by the rest in karyotypic order.

The GRCh37 PAR regions are defined as follows.

Name	Chr	Start	Stop
PAR#1	X	60,001	2,699,520
PAR#2	X	154,931,044	155,260,560
PAR#1	Y	10,001	2,649,520
PAR#2	Y	59,034,050	59,363,566

Isaac Aligner

The Isaac Aligner software uses the following steps to align DNA sequencing data with read lengths of 32–150 bp, single or paired-end, and low error rates.

- **Candidate mapping positions**—Identifies the complete set of relevant candidate mapping positions using a 32-mer seed-based search.
- **Mapping selection**—Selects the best mapping among all candidates.
- **Alignment score**—Determines alignment scores for the selected candidates based on a Bayesian model.

- **Alignment output**—Generates final output in a sorted and duplicate-marked BAM file, realigned indels, and a summary file.

Candidate Mapping

To align reads, the Isaac Aligner first identifies a small but complete set of relevant candidate mapping positions. The Isaac Aligner begins with a seed-based search using 32-mers as seeds. After the initial single-seed search, Isaac performs a multi-seed search for only those reads that were not mapped unambiguously with a single seed.

Mapping Selection

Following a seed-based search, Isaac Aligner selects the best mapping among all candidates.

For paired-end data sets, mappings where only one end is aligned (called orphan mappings) prompt a local search to find additional mapping candidates. These candidates are called shadow mappings. They are defined through the expected minimum and maximum insert size. After optional trimming of low quality 3′ ends and adapter sequences, the possible mapping positions of each fragment are compared. This step accounts for any available pair-end information, possible gaps using a banded Smith-Waterman gap aligner, and possible shadows. The selection is based on the Smith-Waterman score and on the log-probability of each mapping.

Alignment Scores

The alignment scores of each read pair are based on a Bayesian model, where the probability of each mapping is inferred from the base qualities and the positions of the mismatches. The final mapping quality is the alignment score, truncated to 60 for scores above 60, and possibly corrected to known ambiguities in the reference as flagged in the seeds. Following alignment, reads are sorted. Further analysis is performed to identify duplicates and optionally to realign indels.

Alignment Output

After sorting the reads, Isaac Aligner generates compressed binary alignment output files, called BAM (*.bam) files, using the following process:

- **Marking duplicates**—Detection of duplicates is based on the location and observed length of each fragment. The Isaac aligner identifies and marks duplicates even when they appear on oversized fragments or chimeric fragments. Optical duplicates are already filtered out during RTA processing.
- **Realigning indels**—Isaac Aligner tracks previously detected indels, over a window large enough for the current read length, and applies the known indels to all reads with mismatches.
- **Generating BAM files**—The first step in the BAM file generation is the creation of the BAM record, which contains all the required information except the name of the read. The Isaac Aligner reads data from base call (BCL) files that were written during base calling on the sequencer to generate the read names. Data is then compressed into blocks of 64kb or less to create the BAM file.

Strelka (Small Variant Caller)

The Isaac Variant Caller identifies polymorphisms (SNPs) and small indels using the following steps:

- **Read filtering**—Filters out reads failing quality checks.
- **Indel candidate discovery and realignment**—Finds possible indels present in multiple reads and realigns all reads overlapping these candidates.

- SNV calling—Computes the probability of each possible genotype given the aligned read data and a prior distribution of variation in the genome.
- Short range phasing—SNPs within 2 bases of each other, and therefore close enough to be in a single codon, are combined into a single, phased block substitution when read evidence indicates the presence of a consistent diploid solution.
- Indel calling—Analagous to SNV calling, but used for candidate indels.
- Variant rescoring—Assigns final confidence scores based on empirically-fitted models.

Read Filtering

Input reads are filtered by removing any of the following reads:

- Reads that failed base calling quality checks.
- Reads marked as PCR duplicates.
- Paired-end reads not marked as a proper pair.
- Reads with a mapping quality < 20.

Indel Candidate Discovery and Realignment

The variant caller proceeds with candidate indel discovery and generates alternate read alignments based on candidate indels. During realignment, the variant caller selects a representative alignment to use for site genotype calling and depth summarizing by the SNV caller.

SNV Calling

The variant caller filters the set of filtered and realigned reads for SNV calling without affecting indel calls. Any contiguous trailing sequence of N base calls is trimmed from the end of the read. Using a mismatch density filter, reads with an unexpectedly high number of disagreements with the reference are masked as follows.

- The variant caller identifies each insertion or deletion as a single mismatch.
- Base calls with more than 2 mismatches to the reference sequence within 20 bases of the call are ignored.
- If the call occurs within the first or last 20 bases of a read, the mismatch limit is applied to a 41-base window at the corresponding end of the read.
- The mismatch limit is applied to the entire read when the read length is 41 or shorter.

The variant caller filters out all bases marked by the mismatch density filter and any N base calls that remain after the end-trimming step. These filtered base calls are not used for site-genotyping, but appear in the filtered base call counts in the variant caller output for each site.

All remaining base calls are used for site-genotyping. The genotyping method heuristically adjusts the joint error probability that is calculated from multiple observations of the same allele on each strand of the genome. This correction accounts for the possibility of error dependencies.

This method treats the highest-quality base call from each allele and strand as an independent observation and leaves the associated base call quality scores unmodified. Quality scores for subsequent base calls for each allele and strand are then adjusted. This adjustment is done to increase the joint error probability of the given allele above the error expected from independent base call observations.

Short Range Phasing

SNPs within 2 bases of each other, and therefore close enough to be in a single codon, are subject to a postprocessing step. This step can merge them into block substitutions that specify the phasing of the original variants.

Blocks of 2 or more heterozygous variants such that adjacent pairs are within 2 bases of each other are identified. Reads fully spanning a given block are used to score possible haplotype pairs, and if 1 pair of haplotypes is superior to all other alternatives, this pair is output as a block substitution. If, instead, no single pair is clearly best, the variants are output as the original individual calls but with the specification HaplotypeConsistency in the FILTER field.

Indel Calling

Indel candidates are used to score possible indel genotypes similar to the process described for SNVs. Unlike SNVs, there is no correlated error model (ie, reads are treated as fully independent). Indel error probabilities are assigned based on the length of homopolymer runs in the reference and the hypothesized genome implied by an indel candidate.

Variant Rescoring and Filtering

A final calibrated confidence score (GQX) is computed for most variant calls, and this score is used to filter dubious calls. The calibrated score is based on an empirical model fitted to a reference truth set from the Platinum Genomes project. Predictor features are determined for each variant call (including depth of coverage, strand bias, genotype likelihood, mapping, and base qualities). Features are normalized according to average sequencing depth per chromosome and combined in a logistic regression model, to derive a final Q-score. This Q-score is then compared against precomputed cutoffs chosen to balance precision and recall for a separate reference truth set, to determine whether the variant is reported as PASS or filtered.

Variant Call Output

After the SNV and indel genotyping methods and variant rescoring are complete, the variant caller applies a final set of heuristic filters and merges invariant positions with similar properties (depth of coverage, confidence score, and so on) into block records. Then the variant caller reconciles certain conflicts arising when indels overlap other indels or SNVs.

The final output is in the form of a genome variant call (gVCF) file.

gVCF (Genome VCF)

Human genome sequencing applications require sequencing information for both variant and nonvariant positions, yet there is no common exchange format for such data. gVCF addresses this issue.

gVCF is a set of conventions applied to the standard variant call format (VCF). These conventions allow representation of genotype, annotation, and additional information across all sites in the genome, in a reasonably compact format. Typical human whole-genome sequencing results expressed in gVCF with annotation are less than 1.7 GB, or about 1/50 the size of the BAM file used for variant calling.

gVCF is also equally appropriate for representing and compressing targeted sequencing results. Compression is achieved by joining contiguous nonvariant regions with similar properties into single block VCF records. To maximize the utility of gVCF, especially for high stringency applications, the properties of the compressed blocks are conservative. Block properties such as depth and genotype quality reflect the minimum of any site in the block. The gVCF file is also a valid VCF v4.1 file, and can be indexed and used with existing VCF tools such as tabix and IGV. This feature makes the file convenient both for direct interpretation and as a starting point for further analysis.

gvcftools

Illumina has created a full set of utilities aimed at creating and analyzing Genome VCF files. For information and downloads, visit the gvcftools website at sites.google.com/site/gvcftools/home.

Examples

The following is a segment of a VCF file following the gVCF conventions for representation of nonvariant sites and, more specifically, using gvcftools block compression and filtration levels.

In the following gVCF example, nonvariant regions are shown in normal text and variants are shown in **bold**.

NOTE

The variant lines can be extracted from a gVCF file to produce a conventional variant VCF file.

```
20 676194 . A . . PASS END=676441;BLOCKAVG_min30p3a GT:GQX:DP:DPF
   0/0:74:26:0
20 676442 . T G 175 PASS
   SNVHPOL=3;MQ=60;AA=T;GMAF=G|0.255;AF1000G=0.254992;phyloP=-0.016
   GT:GQ:GQX:DP:DPF:AD:ADF:ADR:SB:FT:PL 0/1:208:60:45:3:27,18:16,11:11,7:-
   21:PASS:210,0,325
20 676443 . T . . PASS END=676551;BLOCKAVG_min30p3a GT:GQX:DP:DPF
   0/0:83:29:0
20 676552 . A . . PASS END=676574;BLOCKAVG_min30p3a GT:GQX:DP:DPF
   0/0:51:23:2
20 676575 . A T 377 PASS
   SNVHPOL=3;MQ=60;AA=T;GMAF=A|0.1102;AF1000G=0.889776;phyloP=-0.505
   GT:GQ:GQX:DP:DPF:AD:ADF:ADR:SB:FT:PL 1/1:69:60:24:4:0,24:0,10:0,14:-
   41.9:PASS:370,72,0
20 676576 . T . . PASS END=676672;BLOCKAVG_min30p3a GT:GQX:DP:DPF
   0/0:52:24:1
20 676673 . T . . PASS END=676803;BLOCKAVG_min30p3a GT:GQX:DP:DPF
   0/0:78:27:0
20 676804 . G A 90 PASS
   SNVHPOL=2;MQ=60;AA=G;GMAF=A|0.2526;AF1000G=0.252596;phyloP=-0.066
   GT:GQ:GQX:DP:DPF:AD:ADF:ADR:SB:FT:PL 0/1:123:60:33:1:24,9:11,4:13,5:-
   12.9:PASS:125,0,304
20 676805 . C . . LowGQX;HighDPFRatio . GT:GQX:DP:DPF .:.:0:1
```

In addition to the nonvariant and variant regions in the example, there is also 1 nonvariant region from [676805,676806] that is filtered out due to insufficient confidence that the region is a homozygous reference.

Conventions

Any VCF file following the gVCF convention combines information on variant calls (SNVs and small indels) with genotype and read depth information for all nonvariant positions in the reference. Because this information is integrated into a single file, distinguishing variant, reference, and no-call states for any site of interest is straightforward.

The following subsections describe the general conventions followed in any gVCF file, and provide information on the specific parameters and filters used in the Isaac workflow gVCF output.

NOTE
gVCF conventions are written with the assumption that only one sample per file is being represented.

Interpretation

gVCF files can be interpreted as follows:

- Fast interpretation—As a discrete classification of the genome into variant, reference, and no-call loci. This classification is the simplest way to use the gVCF. The Filter fields for the gVCF file have already been set to mark uncertain calls as filtered for both variant and nonvariant positions. Simple analysis can be performed to look for all loci with a filter value of PASS, and treat them as called.
- Research interpretation—As a statistical genome. Additional fields, such as genotype quality, are provided for both variant and reference positions to allow the threshold between called and uncalled sites to be varied. These fields can also be used to apply more stringent criteria to a set of loci from an initial screen.

External Tools

gVCF is written to the VCF 4.1 specification, so any tool that is compatible with the specification (such as IGV and tabix) can use the file. However, certain tools are not appropriate if they:

- Apply algorithms to VCF files that make sense for only variant calls (as opposed to variant and nonvariant regions in the full gVCF)
- Are only computationally feasible for variant calls

For these cases, extract the variant calls from the full gVCF file.

Special Handling for Indel Conflicts

Sites that are filled in inside deletions have additional treatment.

- Heterozygous Deletions—Sites inside heterozygous deletions have haploid genotype entries (ie, 0 instead of 0/0, 1 instead of 1/1). Heterozygous SNVs are marked with the SiteConflict filter and their original genotype is left unchanged. Sites inside heterozygous deletions cannot have a genotype quality score higher than the enclosing deletion genotype quality.
- Homozygous Deletions—Sites inside homozygous deletions have genotype set to period (.), and site and genotype quality are also set to period (.).
- All Deletions—Sites inside any deletion are marked with the filters of the deletion, and more filters can be added pertaining to the site itself. These modifications reflect the idea that the enclosing indel confidence bounds the site confidence.
- Indel Conflicts—In any region where overlapping deletion evidence cannot be resolved into 2 haplotypes, all indel and set records in the region are marked with the IndelConflict filter.

Table 1 Indel Conflict Filters

ID	Type	Description
IndelConflict	site/indel	Locus is in region with conflicting indel calls.
SiteConflict	site	Site genotype conflicts with proximal indel call. This conflict is typically a heterozygous genotype found inside a heterozygous deletion.

Representation of Nonvariant Segments

This section includes the following subsections:

- Block representation using END key
- Joining nonvariant sites into a single block record
- Block sample values
- Nonvariant block implementations

Block Representation Using END Key

Continuous nonvariant segments of the genome can be represented as single records in gVCF. These records use the standard END key to indicate the extent of the record. Even though the record can span multiple bases, only the first base is provided in the REF field (to reduce file size). Following is a simplified example of a nonreference block record:

```
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the
   variant described in this record">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA19238
chr1 51845 . A . . PASS END=51862
```

The example record spans positions [51845,51862].

Joining Nonvariant Sites Into a Single Block Record

Address the following issues when joining adjacent nonvariant sites into block records:

- The criteria that allow adjacent sites to be joined into a single block record.
- The method to summarize the distribution of SAMPLE or INFO values from each site in the block record.

At any gVCF compression level, a set of sites can be joined into a block if they meet the following criteria:

- Each site is nonvariant with the same genotype call. Expected nonvariant genotype calls are {0/0, 0, ./., .}.
- Each site has the same coverage state, where coverage state refers to whether at least 1 read maps to the site. For example, sites with 0 coverage cannot be joined into the same block with covered sites.
- Each site has the same set of FILTER tags.
- Sites have less than a threshold fraction of nonreference allele observations compared to all observed alleles (based on AD and DP field information). This threshold is used to keep sites with high ratios of nonreference alleles from being compressed into nonvariant blocks. In the Isaac Variant Caller gVCF output, the maximum nonreference fraction is 0.2.

Block Sample Values

Any field provided for a block of sites, such as read depth (using the DP key), shows the minimum observed value among all sites encompassed by the block.

Nonvariant Block Implementations

Files conforming to the gVCF conventions delineated in this document can use different criteria for creation of block records, depending on the desired trade-off between compression and nonvariant site detail. The Isaac Variant Caller provides the blocking scheme min30p3a as the nonvariant block compression scheme.

Each sample value shown for the block, such as the depth (using the DP key), is restricted to have a range where the maximum value is within 30% or 3 of the minimum. Therefore, for sample value range [x,y], $y \leq x+\max(3, x*0.3)$. This range restriction applies to all sample values written in the final block record.

Genotype Quality for Variant and Nonvariant Sites

The gVCF file uses an adapted version of genotype quality for variant and nonvariant site filtration. This value is associated with the GQX key. The GQX value is intended to represent the minimum of Phred genotype quality (assuming the site is variant, assuming the sites is nonvariant).

You can use this value to allow a single value to be used as the primary quality filter for both variant and nonvariant sites. Filtering on this value corresponds to a conservative assumption appropriate for applications where reference genotype calls must be determined at the same stringency as variant genotypes, for example:

- An assertion that a site is homozygous reference at GQX ≥ 30 is made assuming the site is variant.
- An assertion that a site is a nonreference genotype at GQX ≥ 30 is made assuming the site is nonvariant.

Filter Criteria

The gVCF FILTER description is divided into 2 sections, the first describes filtering based on genotype quality while the second describes all other filters.

NOTE

These filters are default values used in the current Isaac Variant Caller implementation. However, no set of filters or cutoff values are required for a file to conform to gVCF conventions.

The genotype quality is the primary filter for all sites in the genome. In particular, traditional discovery-based site quality values that convey confidence that the site is anything besides the homozygous reference genotype, such as SNP or quality, are not used. Instead, a site, or locus, is filtered based on the confidence in the reported genotype for the current sample.

The genotype quality used in gVCF is a Phred-scaled probability that the given genotype is correct. It is indicated with the FORMAT field tag GQX. Any locus where the genotype quality is below the cutoff threshold is filtered with the tag LowGQX. Besides *filtering on genotype quality*, other filters can also be applied.

For more information, see the small variants and genome VCF *Variations* on page 6.

Illumina Annotation Pipeline

Some annotations are provided in the INFO fields of the supplied VCF files. The majority of annotations are provided in the JSON output file described in the section that follows. Illumina compiles annotation content from several sources. SNPs and indels are annotated with the following:

Source	Version	Release Date
dbSNP	147	04/14/2016
COSMIC	v78	05/09/2016
1000 Genomes Project	Phase 3 v5a	05/27/2013
EVS	ESP6500SI-V2-SSA137	11/13/2013
ExAC	0.3.1	03/14/2016
ClinVar	Unknown	08/31/2016
phyloP	hg19	11/10/2009

In addition, the following annotations are added:

- Consequence predictions on RefSeq and Ensembl transcripts (modeled from VEP)
- Annotations in regulatory elements (modeled from VEP)
- Gene/transcript identifiers and their relationship between RefSeq, Ensembl, HGNC, and known synonyms

Annotation Pipeline Output

The output of the Annotation Pipeline is a VCF and gVCF with some annotation in the INFO field as well as a JSON representation of all annotation and sample information (as extracted from the VCF). A JSON format of the annotated records has been created to help promote parsing of the data in a quick efficient manner.

- <samplebarcode>.vcf.gz
- <samplebarcode>.genome.vcf.gz
- <samplebarcode>.json.gz

VCF Record Example

```
##fileformat=VCFv4.1
##reference=hg19
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA12878
chr1 15274 rs62636497 A T 556 PASS
   AA=g;GMAF=G|0.3472;AF1000G=0.640974;phyloP=-1.531;CSQT=1|DDX11L1|NR_
   046018.2|downstream_gene_variant,1|WASH7P|NR_024540.1|intron_
   variant&non_coding_transcript_variant GT:GQ:GQX:DP:DPF:AD:PL
   1/1:117:41:40:4:0,40:370,120,0
```

JSON Record Example

```
"chromosome": "chr1",
"refAllele": "A",
"position": 15274,
"quality": 556,
"filters": ["PASS"],
"altAlleles": ["T"],
"samples": [{
         "variantFreq": 1,
         "totalDepth": 40,
         "alleleDepths": [0,40],
         "genotype": "1/1",
         "genotypeQuality": 41
   }],
   "variants": [{
            "ancestralAllele": "g",
            "altAllele": "T",
            "refAllele": "A",
            "begin": 15274,
            "chromosome": "chr1",
            "phylop46WayScore": -1.531,
            "dbsnp": ["rs62636497"],
            "end": 15274,
            "globalMinorAllele": "G",
            "gmaf": 0.3472,
            "variantType": "SNV",
            "vid": "1:15274:T",
            "transcripts": {
                     "refSeq": [{
                           "transcript": "NR_046018.2",
                           "geneId": "100287102",
                           "hgnc": "DDX11L1",
                           "consequence": ["downstream_gene_variant"],
                           "isCanonical": true
                     },
                     {
                           "transcript": "NR_024540.1",
                           "geneId": "653635",
                           "hgnc": "WASH7P",
                           "consequence": ["intron_variant",
                           "non_coding_transcript_variant"],
                           "hgvsc": "NR_024540.1:n.1233-236T>A",
                           "isCanonical": true
                     }]
         },
         "oneKgAfr": 0.6369,
                "oneKgAll": 0.640974,
                "oneKgAmr": 0.7205,
                "oneKgEas": 0.5188,
                "oneKgEur": 0.7078,
                "oneKgSas": 0.6472
        }]
```

BAM File Conversion

A large volume of data represents the sequence and corresponding alignments, which are provided in BAM format. There are a few methods to convert BAM into different formats, such as FASTQ files.

Picard Tools FASTQ Extraction

Many pipelines start from FASTQ files. To convert BAM files to FASTQ files using Picard tools, refer to the following example.

```
# Convert bam into read1.fastq and read2.fastq
$java -jar /picard-tools-1.110/SamToFastq.jar INPUT=Example.bam
   FASTQ=Example_R1.fastq SECOND_END_FASTQ=Example_R2.fastq VALIDATION_
   STRINGENCY=SILENT

BAM Size: 79 G
Wall Clock Time: 3 hrs 54 min
```

Optional arguments:

- `RE_REVERSE=true`—Reverts the sequence to the native orientation. Otherwise, all aligned sequence is forward orientation.
- `MAX_RECORDS_IN_RAM=5000000`—Decides the number of reads held in memory and controls total memory usage.

Picard requires large amounts of memory. Picard reads data sequentially, line by line from the BAM file, and stores the reads in memory until both pairs of each read have been read. Memory is reset only when the reads are printed. Every read that does not have adjacent or near adjacent pairs requires more memory. Therefore, sort large BAM files when memory is a limiting factor.

Download Picard Tools at sourceforge.net/projects/picard/files/picard-tools.

SAMtools Sort

SAMtools ensures that paired reads are next to each other. You can save a significant amount of memory by using SAMtools to sort the BAM files by name before running Picard.

- sys 17m56.443s
- sys 178m39.641s

Table 2 Example File Sizes

Size	File
152 G	read1.fastq
152 G	read2.fastq
152 G	read2.sorted.fastq
152 G	read2.sorted.fastq
91 G	sorted.bam

Size	File
8.5 M	sorted.bam.bai
125 G	sorted_by_name.bam

Reads Extraction Using SAMtools Flags

The BAM/SAM format contains a bitwise flag column that contains a hexadecimal, which defines the nature of the read. SAMtools allows you to easily filter on reads based on this flag. There are 12 types of these flags. Using the include (-f) or the exclude (-F) option with flags from SAMtools, you can filter or extract any kind of read from the BAM/SAM file.

To convert the SAMtools flags into a human readable format, input the flag into picard.sourceforge.net/explain-flags.html, or run the following command to output the flags in the coded string format described in the SAMtools manual.

```
$samtools view -X Example.bam
```

The following list includes a few commonly used examples of filtering:

- Extract all reads that are unmapped
  ```
  # -f 4 = include reads which are unmapped
  # command will output all the reads which are not mapped.
  $samtools view -h -f 4 Example.bam
  ```
- Extract reads with unmapped mates
  ```
  # -f 8 = include reads whose mates are not mapped
  # command will output all reads whose mates are not mapped.
  $samtools view -h -f 8 Example.bam
  ```
- Extract an unmapped read with a mapped mate
  ```
  # -f 4 = include reads which are unmapped
  # -F 8 = exclude reads whose mate is not mapped
  # command outputs reads that are unmapped with the corresponding mate
    mapped
  $samtools view -h -f 4 -F8 Example.bam
  ```
- Extract a mapped read with an unmapped mate
  ```
  # -f 8 = include reads whose mate is unmapped
  # -F 8 = exclude all reads not mapped
  # command outputs reads which are mapped with the mate is unmapped
  $samtools view -h -f 8 -F4 Example.bam
  ```
- Extract both reads of a pair, which are unmapped
  ```
  #-f 12 = a combination of flag 4 and flag 8 (4+8) -> include only if a
    read is unmapped and the mate is unmapped.
  # command outputs read pairs with both pairs unmapped
  $samtools view -h -f 12 Example.bam
  ```

Technical Assistance

For technical assistance, contact Illumina Technical Support.

Website:	www.illumina.com
Email:	techsupport@illumina.com

Illumina Customer Support Telephone Numbers

Region	Toll Free	Regional
North America	+1.800.809.4566	
Australia	+1.800.775.688	
Austria	+43 800006249	+43 19286540
Belgium	+32 80077160	+32 34002973
China	400.635.9898	
Denmark	+45 80820183	+45 89871156
Finland	+358 800918363	+358 974790110
France	+33 805102193	+33 170770446
Germany	+49 8001014940	+49 8938035677
Hong Kong	800960230	
Ireland	+353 1800936608	+353 016950506
Italy	+39 800985513	+39 236003759
Japan	0800.111.5011	
Netherlands	+31 8000222493	+31 207132960
New Zealand	0800.451.650	
Norway	+47 800 16836	+47 21939693
Singapore	+1.800.579.2745	
Spain	+34 911899417	+34 800300143
Sweden	+46 850619671	+46 200883979
Switzerland	+41 565800000	+41 800200442
Taiwan	00806651752	
United Kingdom	+44 8000126019	+44 2073057197
Other countries	+44.1799.534000	

Safety data sheets (SDSs)—Available on the Illumina website at support.illumina.com/sds.html.

Clinical Services Assistance

For assistance, contact Illumina Clincial Services Support.

Website:	http://www.illumina.com/clinical/illumina_clinical_laboratory.html
Email:	everygenome@illumina.com
Telephone:	+1.858.736.8080

Illumina
5200 Illumina Way
San Diego, California 92122 U.S.A.
+1.800.809.ILMN (4566)
+1.858.202.4566 (outside North America)
techsupport@illumina.com
www.illumina.com