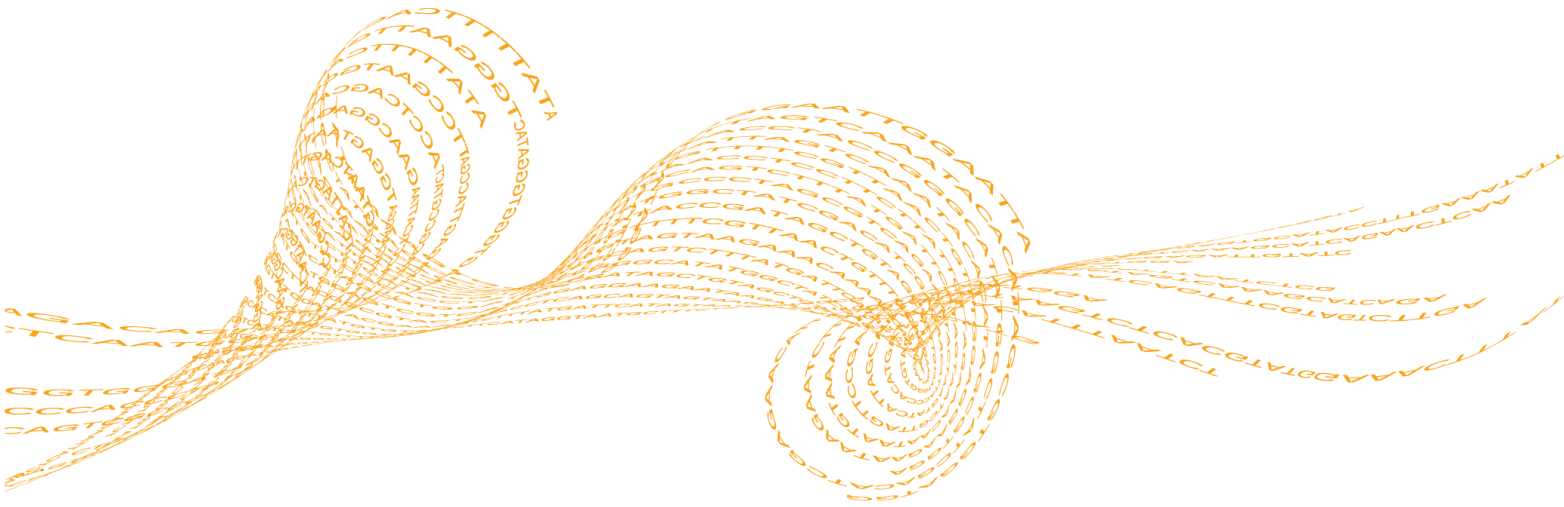
Isaac Whole Genome Sequencing v4

App Guide

For Research Use Only. Not for use in diagnostic procedures.



ILLUMINA PROPRIETARY
15050954 Rev. C
June 2015

illumina®

Introduction

After BaseSpace® has generated FASTQ files containing the base calls and quality scores of a run, you can use the Isaac Whole Genome Sequencing v4 app to analyze the sequencing data. The resulting data are analyzed in 2 steps: alignment to the reference genome followed by assembly and variant calling.

The Isaac Whole Genome Sequencing v4 app uses the following modules for analysis:

- Alignment with the Isaac Alignment Software
- Variant calling with the Isaac Variant Caller
- Structural variant (SV) and large indels with the Isaac SV Caller.
- Copy number variation (CNV) analysis with the Isaac CNV Caller.

These modules produce the following output:

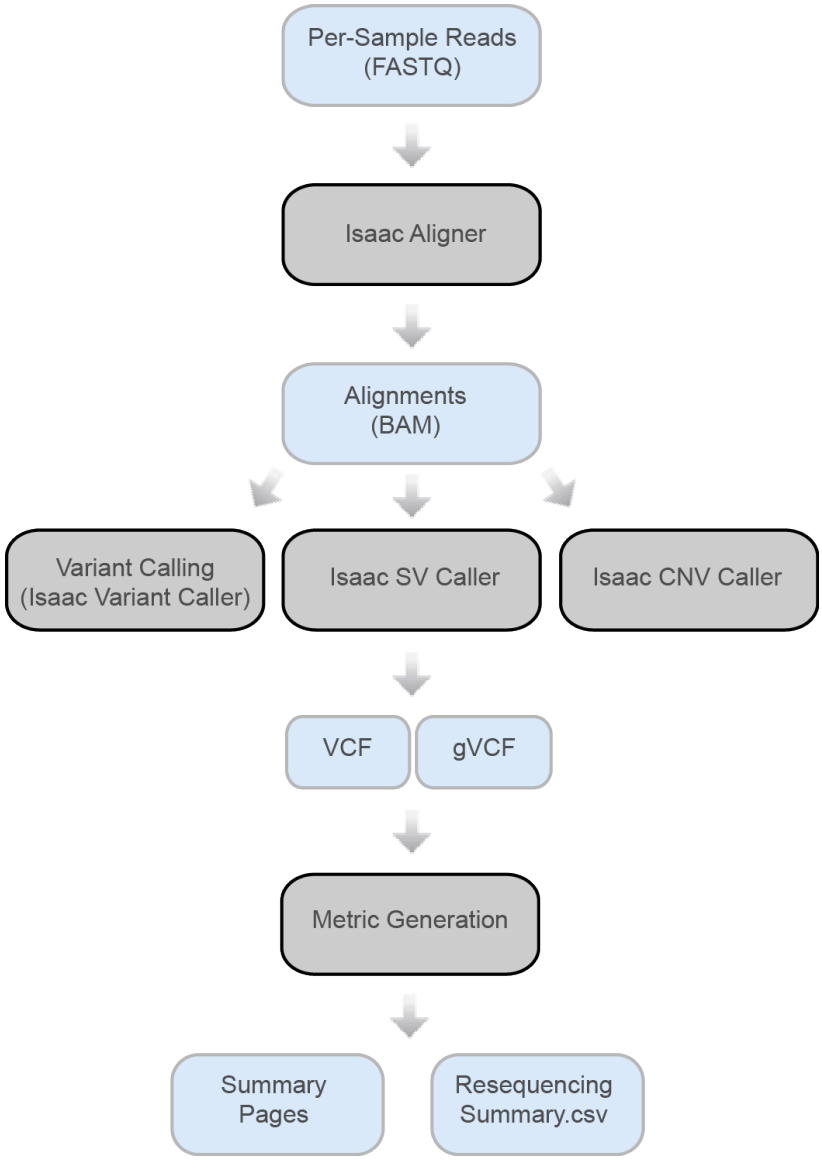
- Realigned and duplicate-marked reads in BAM file format.
- Variants in VCF file format.
- An additional Genome VCF (gVCF) file. This file features an entry for every base in the reference, which differentiates reference calls and no calls, and a summary of quality.

In addition, there is a metric generation step with the following output:

- Summary pages
- Summary report (*.summary.csv)

See *Isaac Whole Genome Sequencing v4 Methods* on page 21 and *Isaac Whole Genome Sequencing v4 Output* on page 7 for more information.

Figure 1 Isaac Whole Genome Sequencing v4 Workflow



Versions

The following module versions are used in the Isaac Whole Genome Sequencing v4 app:

Category	Software	Version
Aligner	Isaac Aligner	SAAC00776.15.05.08
Analysis Software	ISIS	2.5.55.16
Annotation Service	IONA	1.0.11.0
Variant Caller	Isaac Variant Caller (Starling)	2.1.4.2
Variant Caller	Isaac Structural Variant Caller (Manta)	0.23.1
Variant Caller	Isaac Copy Number Variant Caller (Canvas)	1.1.0.5
Other	Samtools	0.1.19-isis-1.0.1
Other	Tabix	0.2.6

Current Limitations

Before running the Isaac Whole Genome Sequencing v4 app, be aware of the following limitations:

- Reference Genomes
  - Human, UCSC hg19
  - Human, Ensembl GRCh37

  The reference genomes are PAR-Masked, which means that the Y chromosome sequence has the Pseudo Autosomal Regions (PAR) masked (set to N) to avoid mismapping of reads in the duplicate regions of sex chromosomes.
- Currently the app does not support mate-pair or other nonforward/reverse styles of paired-end sequencing.
- Requires a minimum read length of 32 bp and a maximum read length of 150 bp.
- Minimum recommended data set size is enough data to yield 10 × coverage after alignment of the genome being sequenced. Recommended minimum number of reads for 10 × coverage with an additional 5% to account for unaligned reads.
  - Genome size—3.3 Gb
  - Data size—35 Gb
  - Reads for 2 x 100—175.00 (million)
  - Reads for 2 x 150—116.67 (million)
- Maximum data set size must be fewer than 240 gigabases, which equates to the following:
  - Approximately 1.2 billion reads assuming 2 × 100
  - Approximately 800 million reads assuming 2 × 150
- CompletedJobInfo.xml may not print all statistics.
- Sample name length has a maximum of 32 characters.

Running Isaac Whole Genome Sequencing v4

1. Navigate to the project or sample that you want to analyze.
2. Click the **Apps** tab and select **Isaac Whole Genome Sequencing v4**.
3. Click **Launch** to open the app.
4. Fill out the required fields in the Isaac Whole Genome Sequencing v4 input form:
   - **Analysis Name**—Provide the analysis name. Default name is the app name with the date and time the app session was started.
   - **Save Results To**—Select the project that stores the app results.
   - **Sample**—Browse to the sample you want to analyze, and select the checkbox.
   - **Reference Genome**—Select the reference genome.
   - **Disable Variant Calling**—If selected, variants, structural variant (SV), and copy number variant (CNV) calling is disabled.
   - **Enable SV calling**—If selected, structural variants calling is performed for paired end-data. If Disable Variant Calling is selected, this option is disabled.
   - **Enable CNV calling**—If selected, copy number variants (CNV) calling is performed. If Disable Variant Calling is selected, this option is disabled.
     For more information, see *Isaac Structural Variant Caller* on page 28 and *Isaac Copy Number Variant Caller* on page 23.
5. [Optional] Open the **Advanced** menu and select **Flag PCR Duplicates**. If selected, PCR duplicates are flagged in the BAM files and not used for variant calling. PCR duplicates are defined as paired-end reads generated from 2 clusters that have the exact same alignment positions for each read. Optical duplicates are filtered out during RTA processing.
6. Click **Continue**.

The Isaac Whole Genome Sequencing v4 app starts analyzing your sample. When completed, the status of the app session is automatically updated, and you receive an email.

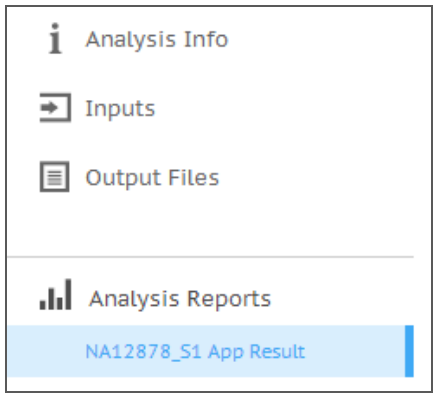
NOTE

If needed, you can merge sampleFASTQ files in BaseSpace. See the *BaseSpace User Guide (part # 15044182)* for more information.

Isaac Whole Genome Sequencing v4 Output

To view the output, click the **Projects** tab, then the project name, and then the analysis.

Figure 2 Isaac Whole Genome Sequencing v4 Output Navigation Bar



When the App Session is completed, you can access your output through the left navigation bar.

- **Analysis Info**—Overview of the app session settings. See *Analysis Info* on page 7.
- **Inputs**—Overview of input settings, see *Inputs* on page 8.
- **Output Files**—Access to output files, organized by sample and app session. See *Output Files* on page 8.
- **Analysis Reports**—Access to analysis metrics for each sample. See *Analysis Reports* on page 17.

Analysis Info

This app provides an overview of the analysis on the Analysis Info page.

A brief description of the metrics is below.

Row	Definition
Name	Name of the app session.
Application	App that generated this analysis.
Date Started	Date and time the app session started.
Date Completed	Date and time the app session completed.
Duration	Duration of analysis.
Session Type	The number of nodes used.
Size	Total size of all output files.
Status	Status of the app session.

Log Files

Click the **Log Files** link on the Analysis Info page to access the app log files.

The key log files to help follow data processing and debugging are the following:

- **CompletedJobInfo.xml**—Contains information about the completed job.

- **Logging.zip**—Contains all detailed workflow log files for each step of the workflow.
- **output-{timestamp}.log**—Shows the raw console output from the app.
- **ResequencingRunStatistics.xml**—Contains statistics about the completed job.
- **SampleSheet.csv**—Sample sheet.
- **SampleSheetUsed.csv**—A copy of the sample sheet, generated at the end of a run.
- **spacedock-{timestamp}.log**—Internal application log file not for general customer usage.
- **spacedock-infrastructure-{timestamp}.log**—Internal application log file not for general customer usage.
- **uploader-{timestamp}.log**—Internal application log file not for general customer usage.
- **UsageLog.txt**—Shows system resources logging information.

Isaac Whole Genome Sequencing v4 Status

The status of the Isaac Whole Genome Sequencing v4 app session can have the following values:

1. Launching Isis
2. Alignment
3. Detect CNV (only if SV/CNV detection option is selected from input form)
4. Detect SV (only if SV/CNV detection option is selected from input form)
5. Variant analysis
6. Statistics evaluation
7. Report generation

Inputs

The Inputs page provides an overview of the input samples and settings that were specified when the Isaac Whole Genome Sequencing v4 analysis was set up.

Output Files

The Output Files page provides access to the output files.

BAM Files

The Sequence Alignment/Map (SAM) format is a generic alignment format for storing read alignments against reference sequences, supporting short and long reads (up to 128 Mb) produced by different sequencing platforms. SAM is a text file format that is human-readable. The Binary Alignment/Map (BAM) keeps the same information as SAM, but in a compressed, binary format that is only machine readable.

Detailed Description

The file naming convention for aligned reads in BAM format is as follows: SampleName_S#.bam (where # is the sample number determined by ordering in the sample sheet).

Go to samtools.sourceforge.net/SAM1.pdf to see the exact SAM specification.

BWA adds some custom fields to the BAM output. See bio-bwa.sourceforge.net/bwa.shtml#4 for a description.

VCF Files

VCF is a text file format that contains information about variants found at specific positions in a reference genome. The file format consists of meta-information lines, a header line, and then data lines. Each data line contains information about a single variant.

More information is available here: www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41.

VCF File Format

The file naming convention for VCF files is as follows: SampleName_S#.vcf (where # is the sample number determined by ordering in the sample sheet).

The header of the VCF file describes the tags used in the remainder of the file. A description of the tags is also provided here and on www.broadinstitute.org/gatk/guide/article?id=1268.

Setting	Description
CHROM	The chromosome of the reference genome. Chromosomes appear in the same order as the reference FASTA file (generally karyotype order)
POS	The 1-based position of this variant in the reference chromosome. The convention for *.vcf files is that, for SNPs, this base is the reference base with the variant. For indels or deletions, this base is the reference base **immediately before** the variant. Variants are ordered by position.
ID	The rs number for the SNP obtained from dbSNP. If there are multiple rs numbers at this location, the list is semi-colon delimited. If no dbSNP entry exists at this position, the missing value ('.') is used.
REF	The reference genotype. For example, a deletion of a single T can be represented as reference TT and alternate T.
ALT	The alleles that differ from the reference read. For example, an insertion of a single T can be represented as reference A and alternate AT.
QUAL	A Phred-scaled quality score assigned by the variant caller. Higher scores indicate higher confidence in the variant (and lower probability of errors). For a quality score of Q, the estimated probability of an error is 10-(Q/10). For example, the set of Q30 calls has a 0.1% error rate. Many variant callers assign quality scores (based on their statistical models) which are high relative to the error rate observed in practice.
FILTER	See *VCF FILTER Entries* on page 10 for possible entries.
FORMAT	See *VCF FORMAT Entries* on page 10 for possible entries.

Setting	Description
INFO	See *VCF INFO Entries* on page 10 for possible entries.
SAMPLE	The sample column gives the values specified in the FORMAT column. One MAXGT sample column is provided for the normal genotyping (assuming the reference). For reference, a second column is provided for genotyping assuming the site is polymorphic.

Isaac Whole Genome Sequencing v4 VCF Entries

The VCF files for Isaac Whole Genome Sequencing v4 can have the following entries in the FILTER, FORMAT, and INFO fields:

Table 1 VCF FILTER Entries

Entry	Description
IndelConflict	Locus is in region with conflicting indel calls
SiteConflict	Site genotype conflicts with proximal indel call, typically a heterozygous SNV call made inside of a heterozygous deletion
LowGQX	Locus GQX is less than 30 or not present
HighDPFRatio	The fraction of base calls filtered out at a site is greater than 0.4
HighSNVSB	SNV strand bias value (SNVSB) exceeds 10

Table 2 VCF FORMAT Entries

Entry	Description
GQX	Minimum of {Genotype quality assuming variant position,Genotype quality assuming non-variant position}
GT	Genotype
GQ	Genotype Quality
DP	Filtered base call depth used for site genotyping
DPF	Base calls filtered from input before site genotyping
AD	Allelic depths for the ref and alt alleles in the order listed. For indels, this value only includes reads that confidently support each allele (posterior probability 0.999 or higher that read contains indicated allele vs all other intersecting indel alleles)
DPI	Read depth associated with indel, taken from the position preceding the indel.

Table 3 VCF INFO Entries

Entry	Description
SNVSB	SNV site strand bias
SNVHPOL	SNV contextual homopolymer length
CIGAR	CIGAR alignment for each alternate indel allele

Entry	Description
RU	Smallest repeating sequence unit extended or contracted in the indel allele relative to the reference. RUs longer than 20 bases are not reported.
REFREP	Number of times RU is repeated in reference.
IDREP	Number of times RU is repeated in indel allele.
END	End position of the region described in this record
BLOCKAVG_min30p3a	Non-variant site block. All sites in a block are constrained to be non-variant, have the same filter value, and have all sample values in range [x,y], y <= max(x+3,(x*1.3)). All printed site block sample values are the minimum observed in the region spanned by the block

Genome VCF (gVCF)

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 up to date 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.

```
chr20 676337 . T . 0.00 PASS END=676401;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:143:51:0
```

```
chr20 676402 . A . 0.00 PASS END=676441;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:169:57:0
chr20 676442 . T G 287.00 PASS SNVSB=-30.5;SNVHPOL=3
   GT:GQ:GQX:DP:DPF:AD 0/1:316:287:66:1:33,33
chr20 676443 . T . 0.00 PASS END=676468;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:202:68:1
chr20 676469 . G . 0.00 PASS . GT:GQX:DP:DPF 0/0:199:67:5
chr20 676470 . A . 0.00 PASS END=676528;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:157:53:0
chr20 676529 . T . 0.00 PASS END=676566;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:120:41:0
chr20 676567 . C . 0.00 PASS END=676574;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:114:39:0
chr20 676575 . A T 555.00 PASS SNVSB=-50.0;SNVHPOL=3
   GT:GQ:GQX:DP:DPF:AD 1/1:114:114:39:0:0,39
chr20 676576 . T . 0.00 PASS END=676625;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:95:36:0
chr20 676626 . T . 0.00 PASS END=676650;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:117:40:0
chr20 676651 . T . 0.00 PASS END=676698;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:90:31:0
chr20 676699 . T . 0.00 PASS END=676728;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:69:24:0
chr20 676729 . C . 0.00 PASS END=676783;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:57:20:0
chr20 676784 . C . 0.00 PASS END=676803;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:51:18:0
chr20 676804 . G A 62.00 PASS SNVSB=-7.5;SNVHPOL=2
   GT:GQ:GQX:DP:DPF:AD 0/1:95:62:17:0:11,66
chr20 676805 . C . 0.00 PASS END=676818;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:48:17:0
chr20 676819 . T . 0.00 PASS END=676824;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:39:14:0
chr20 676825 . A . 0.00 PASS END=676836;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:30:11:0
chr20 676837 . T . 0.00 LowGQX END=676857;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:21:8:0
chr20 676858 . G . 0.00 PASS END=676873;BLOCKAVG_min30p3a
   GT:GQX:DP:DPF 0/0:30:11:0
```

In addition to the nonvariant and variant regions in the example, there is also 1 nonvariant region from [676837,676857] that is filtered out due to insufficient confidence that the region is 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

gVCFs file 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 specifications, 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 variants 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 4 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 Non-Variant 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" INFO 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 Non-Variant 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...

- 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 following 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.

Filter Criteria

The gVCF FILTER description is divided into 2 sections: (1) describes filtering based on genotype quality; (2) 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 SNV 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. In addition to filtering on genotype quality, some other filters are also applied.

The gVCF output from Isaac Variant Caller includes several heuristic filters applied to the site and indel records. The filters are as follows.

Table 5 VCF Site and Indel Record Filters

VCF Filter ID	Type	Description
HAPLOID_ CONFLICT	site/indel	Locus has heterozygous genotype in a haploid region.
HighDepth	site/indel	The locus depth is greater than 3x the mean chromosome depth.
HighDPFRatio	site	The fraction of base calls filtered out at a site is greater than 0.3.
HighSNVSB	site	SNV strand bias value (SNVSB) exceeds 10.
IndelConflict	indel	The locus is in region with conflicting indel calls.
IndelSizeFilter	indel	Indel is outside reportable size range. Insertion/Deletion range reported in VCF header.
LowGQX	site/indel	Locus GQX is less than 30 or not present.
SiteConflict	indel	The site genotype conflicts with the proximal indel call. This call is typically a heterozygous SNV call made inside a heterozygous deletion.

Summary Report (*.summary.csv)

The Isaac Whole Genome Sequencing v4 app generates a Resequencing Summary Report containing sample results in a comma-separated values (CSV) format (*.summary.csv). This report is an overview of statistics for the sample.

Statistic	Definition
Sample ID	IDs of samples reported on in the file.
Run Folder	Run folders for samples reported on in the file.
Reference Genome	The genome and version selected.
Percent Duplicate Paired Reads	Percentage of paired reads that have duplicates.
Fragment length median	Median length of the sequenced fragment. The fragment length is calculated based on the locations at which a read pair aligns to the reference. The read mapping information is parsed from the BAM files.
Fragment length min	Minimum length of the sequenced fragment.
Fragment length max	Maximum length of the sequenced fragment.
Fragment length SD	Standard deviation of the sequenced fragment length.
Percent Aligned (per read)	The percentage of reads passing filter that aligned to the reference genome.
Percent Q30 (per read)	The percentage of bases with a quality score of 30 or higher.
MismatchRate (per read)	The average percentage of mismatches across both reads 1 and 2 over all cycles.
SNVs All	Total number of Single Nucleotide Variants present in the data set passing the quality filters.
SNVs	SNVs passing variants filter.
SNVs (Percent found in dbSNP)	100*(Number of SNVs in dbSNP/Number of SNVs). The SNVs that were found in the dbSNP are annotated accordingly.
SNV Ts/Tv ratio	The number of Transition SNVs that pass the quality filters divided by the number of Transversion SNVs that pass the quality filters. Transitions are interchanges of purines (A, G) or of pyrimidines (C, T). Transversions are interchanges of purine and pyrimidine bases (for example, A to T).
SNV Het/Hom ratio	Number of heterozygous SNVs/Number of homozygous SNVs.
Indels	Total number of indels present in the data set passing the quality filters.
Insertions	Insertions passing variant filters.
Deletions	Deletions passing variant filters.
Indels (Percent found in dbSNP)	100*(Number of Indels in dbSNP/Number of Indels).

Statistic	Definition
Insertions (Percent found in dbSNP)	100*(Number of insertions in dbSNP/ Number of insertions)
Deletions(Percent found in dbSNP)	100*(Number of deletions in dbSNP/ Number of deletions)
Indel Het/Hom ratio	Number of heterozygous indels/Number of homozygous indels.
Insertion Het/Hom ratio	Ratio of the number of heterozygous to homozygous insertions.
Deletion Het/Hom ratio	Ratio of the number of heterozygous to homozygous deletions.
SmallVariantStatisticsFlag	Flags whether SmallVariantStatistics was run (1 means that it was run)
TotalNumber[variant]	Variants passing filter.
Num[variant]InGenes	The number of variants that fall into a gene.
Num[variants]InExons	The number of variants that fall into an exon.
Num[variants] InCodingRegions	The number of variants that fall into a coding region.
SpliceSiteRegion [variants]	The number of variants that fall into a splice site region.
StopGained[variants]	The number of variants that cause an additional stop codon.
StopLost[variants]	The number of variants that cause the loss of a stop codon.
FrameShift[variants]	The number of variants that cause a frameshift.
NonSynonymous [variants]	The number of variants that cause an amino acid change in a coding region.
Synonymous[variants]	The number of variants that are within a coding region, but do not cause an amino acid change.
MatureMiRNA[variants]	The number of variants that fall into a mature microRNA.
UTRRegion[variants]	The number of variants that fall into an untranslated region (UTR).
SVStatisticsFlag	Flags whether SVStatistics was run (1 means that it was run)
CNVStatisticsFlag	Flags whether CNVStatistics was run (1 means that it was run)

Sample Report

The Sample Report presents the same data as the Analysis report, but as download (HTML or PDF). These files are located in the results folder.

For a description of the presented metrics, see *Analysis Reports* on page 17.

Analysis Reports

The Isaac Whole Genome Sequencing v4 app provides an overview of statistics per sample on the Analysis Reports sample pages.

Alignment Summary

Statistic	Definition
Number of Reads	Total number of reads passing filter for this sample.
Coverage	Total number of aligned bases divided by the genome size.
Percent Duplicate Paired Reads	Percentage of paired reads that have duplicates.
Fragment Length Median	Median length of the sequenced fragment. The fragment length is calculated based on the locations at which a read pair aligns to the reference. The read mapping information is parsed from the BAM files.
Fragment Length Standard Deviation	Standard deviation of the sequenced fragment length.

▸ Read Statistics

Statistic	Definition
Percent Aligned	The percentage of reads passing filter that aligned to the reference genome.
Percent Q30	The percentage of bases with a quality score of 30 or higher.
Mismatch Rate	The average percentage of mismatches across both reads 1 and 2 over all cycles.

Small Variants Summary

This table provides metrics about the number of SNVs, insertions, and deletions.

Statistic	Definition
Total Passing	The total number of variants present in the data set that passed the variant quality filters.
Percent Found in dbSNP	100*(Number of variants in dbSNP/Number of variants).
Het/Hom Ratio	Number of heterozygous variants/Number of homozygous variants.
Ts/Tv Ratio	Transition rate of SNVs that pass the quality filters divided by transversion rate of SNVs that pass the quality filters. Transitions are interchanges of purines (A, G) or of pyrimidines (C, T). Transversions are interchanges between purine and pyrimidine bases (for example, A to T).

Variants by Sequence Context

Statistic	Definition
Number in Genes	The number of variants that fall into a gene.
Number in Exons	The number of variants that fall into an exon.
Number in Coding Regions	The number of variants that fall into a coding region.
Number in UTR Regions	The number of variants that fall into an untranslated region (UTR).
Number in Mature microRNA	The number of variants that fall into a mature microRNA.
Number in Splice Site Regions	The number of variants that fall into a splice site region.

Variants by Consequence

Statistic	Definition
Frameshifts	The number of variants that cause a frameshift.
Non-synonymous	The number of variants that cause an amino acid change in a coding region.
Synonymous	The number of variants that are within a coding region, but do not cause an amino acid change.
Stop Gained	The number of variants that cause an additional stop codon.
Stop Lost	The number of variants that cause the loss of a stop codon.

Structural Variants Summary

This table breaks structural variant output into the classes of variants called, and reports the total number and their overlap with annotated genes. All counts are based on PASS filter variants.

Variant Class	Definition
CNV	A copy-number variation (CNV) is a large category of structural variation, which includes insertions, deletions and duplications. CNVs are generally greater than 10 kb. CNVs below 10 kb are filtered but still present in the VCF file.
Insertion	In an insertion, nucleotides are added between two adjacent nucleotides in the reference sequence. The insertions in the structural variants category are 51 bp or greater.
Tandem Duplication	In a tandem duplication, a segment of a chromosome is duplicated front to end, with both segments in the same orientation. The segments are 51 bp or greater.
Deletion	In a deletion, contiguous nucleotides are absent compared to the reference sequence. The deletions in the structural variants category are 51 bp or greater.

Variant Class	Definition
Inversion	An inversion is a chromosome rearrangement in which a segment of a chromosome is reversed end to end. An inversion occurs when a single chromosome undergoes breakage and rearrangement within itself. The segments are 51 bp or greater.

Coverage Histogram

The coverage histogram shows the number of reference bases plotted against the depth of coverage (read depth). It has the following features:

- The dropdown menu allows you to look at the overall picture, or highlight a particular chromosome.
- The **Fix Y Scale** checkbox allows you to keep the Y Scale the same when comparing multiple chromosomes.
- The **Export TSV** link allows you to export the coverage data in a tab-separated TXT file.

Isaac Whole Genome Sequencing v4 Methods

This chapter describes the methods that are used in the Isaac Whole Genome Sequencing v4 app.

Isaac Aligner

The Isaac Aligner aligns DNA sequencing data, single or paired-end, with read lengths 32–150 bp and low error rates using the following steps:

- **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 duplicate-marked BAM file, and summary file.

1 Come Raczy, Roman Petrovski, Christopher T. Saunders, Ilya Chorny, Semyon Kruglyak, Elliott H. Margulies, Han-Yu Chuang, Morten Källberg, Swathi A. Kumar, Arnold Liao, Kristina M. Little, Michael P. Strömberg and Stephen W. Tanner (2013) Isaac: Ultra-fast whole genome secondary analysis on Illumina sequencing platforms. Bioinformatics 29(16):2041-3 bioinformatics.oxfordjournals.org/content/29/16/2041

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 from the extremities of the read as seeds. Isaac Aligner performs another search using different seeds for only those reads that were not mapped unambiguously with the first pass seeds.

Mapping Selection

Following a seed-based search, the Isaac Aligner selects the best mapping among all the candidates. For paired-end data sets, all mappings where only one end is aligned (called orphan mappings) trigger a local search to find additional mapping candidates. These candidates (called shadow mappings) 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 takes into account pair-end information (when available), 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 (MAPQ) is the alignment score, truncated to 60 for scores above 60, and corrected based on known ambiguities in the reference flagged during candidate mapping. Following alignment, reads are sorted. Further analysis is performed to identify duplicates and optionally to realign indels.

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. Following alignment, reads are sorted. Further analysis is performed to identify duplicates and optionally to realign indels.

Alignment Output

After sorting the reads, the 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.
- **Realigning indels**—The 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 BAM file generation is creation of the BAM record, which contains all 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 are then compressed into blocks of 64 kb or less to create the BAM file.

Isaac Variant Caller

The Isaac Variant Caller identifies single nucleotide variants (SNVs) and small indels using the following steps:

- **Read filtering**—Filters out reads failing quality checks.
- **Indel calling**—Identifies a set of possible indel candidates and realigns all reads overlapping the candidates using a multiple sequence aligner.
- **SNV calling**—Computes the probability of each possible genotype given the aligned read data and a prior distribution of variation in the genome.
- **Indel genotypes**—Calls indel genotypes and assigns probabilities.
- **Variant call output**—Generates output in a VCF file and a compressed genome variant call (gVCF) file. See *VCF Files* on page 9 and *Genome VCF (gVCF)* on page 11 for details.

Indel Candidates

Input reads are filtered by removing any of the following:

- 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 less than 20.

Indel Calling

The variant caller proceeds with candidate indel discovery and generates alternate read alignments based on the candidate indels. As part of the realignment process, the variant caller selects a representative alignment to be used for site genotype calling and depth summarization by the SNV caller.

SNV Calling

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

- The variant caller treats 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.

Indel Genotypes

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.

Variant Call Output

After the SNV and indel genotyping methods are complete, the variant caller applies a final set of heuristic filters to produce the final set of calls in the output.

The output in the genome variant call (gVCF) file captures the genotype at each position and the probability that the consensus call differs from reference. This score is expressed as a Phred-scaled quality score.

Isaac Copy Number Variant Caller

Isaac Copy Number Variant (CNV) Caller is an algorithm for calling copy number variants from a diploid sample. Most of a normal DNA sample is diploid, or having 2 copies. Isaac CNV Caller identifies regions of the sample genome that are not present, or present either one time or more than 2 times in the genome. Isaac CNV Caller scans the genome for regions having an unexpected number of short read alignments. Regions with fewer than the expected number of alignments are classified as losses. Regions having more than the expected number of alignments are classified as gains.

Isaac CNV Caller is appropriately applied to low-depth cytogenetics experiments, low-depth single-cell experiments, or whole-genome sequencing experiments. Isaac CNV Caller is not appropriate for whole exome experiments, cancer studies, or any other experiment with the following conditions:

- Most of the genome is not assumed to be diploid.
- Reads are not distributed randomly across the diploid genome.

Workflow

Isaac CNV Caller can be conceptually divided into 4 processes:

- Binning—Counting alignments in genomic bins.
- Cleaning—Removal of systematic biases and outliers from the counts.
- Partitioning—Partitioning the counts into homogenous regions.
- Calling—Assigning a copy number to each homogenous region.

These processes are explained in subsequent sections.

Binning

The binning procedure creates genomic windows, or bins, across the genome and counts the number of observed alignments that fall into each bin. The alignments are provided in the form of a BAM file.

Isaac CNV Caller binning keeps in memory a collection of BitArrays to store observed alignments, one BitArray for each chromosome. Each BitArray length is the same as its corresponding chromosome length. As the BAM file is read in, Isaac CNV Caller records the position of the left-most base in each alignment within the chromosome-appropriate BitArray. After all alignments in the BAM file have been read, the BitArrays have a "1" wherever an alignment was observed and a "0" everywhere else.

After reading in the BAM file, a masked FASTA file is read in, one chromosome at a time. This FASTA file contains the genomic sequences that were used for alignment. Each 35-mer within this FASTA file is marked as unique or nonunique with uppercase and lowercase letters. If a 35-mer is unique, then its first nucleotide is capitalized; otherwise, it is not capitalized. For example, in the sequence:

```
acgtttaATgacgatGaacgatcagctaagaatacgacaatatcagacaa
```

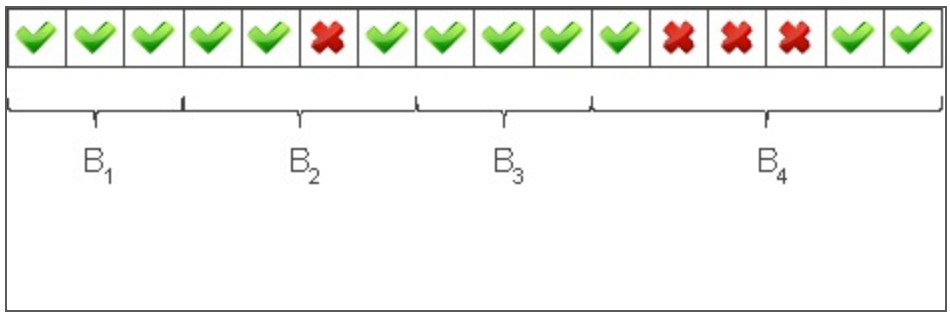
The 35-mers marked as unique are as follows:

```
ATGACGATGAACGATCAGCTAAGAATACGACAATA
TGACGATGAACGATCAGCTAAGAATACGACAATAT
GAACGATCAGCTAAGAATACGACAATATCAGACAA
```

Isaac CNV Caller stores the genomic locations of unique 35-mers in another collection of BitArrays analogous to BitArrays used to store alignment positions. Unique positions and nonunique positions are marked with "1"s and "0"s, respectively. This marking is used as a mask to guarantee that only alignments that start at unique 35-mer positions in the genome are used.

Bin Sizes

Isaac CNV Caller is initialized with 100 alignments per bin and then proceeds to compute the bin boundaries such that each bin contains the same bin size, or number of unique 35-mers. The term "bin size" refers to the number of unique genomic 35-mers per bin. Because some regions of the human genome are more repetitive than others, physical bin sizes (in genomic coordinates) are not identical. In the following example, each box is a position along the genome. Each checkmark represents a unique 35-mer while each X represents a nonunique 35-mer. The bin size in this example is 3 (3 checkmarks per bin). The physical size of each bin is not constant. B1 and B3 have a physical size of 3 but B2 and B4 have physical sizes of 4 and 6, respectively.



Computing Bin Size

To compute bin size, the ratio of observed alignments to unique 35-mers is calculated for each autosome. The desired number of alignments per bin is then divided by the median of these ratios to yield bin size. For whole-genome sequencing, bin sizes are typically in the range of 800–1000 unique 35-mers. Correspondingly, most physical window sizes are in the 1–1.2 kb range. The advantage of this approach relative to using fixed genomic intervals is that the same number of reads map to each bin, regardless of "uniqueness" or ability to be mapped.

After bin size is computed, bins are defined as consecutive genomic windows such that each bin contains the same bin size, or number of unique 35-mers. The number of observed alignments present within the boundary of each bin is then counted from the alignment BitArrays. The GC content of each bin is also calculated. The chromosome, genomic start, genomic stop, observed counts and GC content in each bin are output to disk.

Cleaning

The Isaac CNV Caller cleaning comprises the following 3 procedures that remove outliers and systematic biases from the count data computed in Isaac CNV Caller.

1 Single point outlier removal.

2 Physical size outlier removal.

3 GC content correction.

These procedures are performed on the bins produced during the Isaac CNV Caller binning process.

Single Point Outlier Removal

This step removes individual bins that represent extreme outliers. These bins have counts that are very different from the counts present in upstream and downstream bins. Two values, a and b, are defined as to be very different when their difference is greater than expected by chance, assuming a and b come from the same underlying distribution. These values use the Chi-squared distribution, as follows:

$\mu = 0.5a + 0.5b$

$\chi 2 = ((a - \mu)^2 + (b - \mu)^2)\ \mu\text{-}1$

A value of $\chi 2$ greater than 6.635, which is the 99th percentile of the Chi-squared distribution with 1 degree of freedom, is considered very different. If a bin count is very different from the count of both upstream and downstream neighbors, then the bin is deemed an outlier and removed.

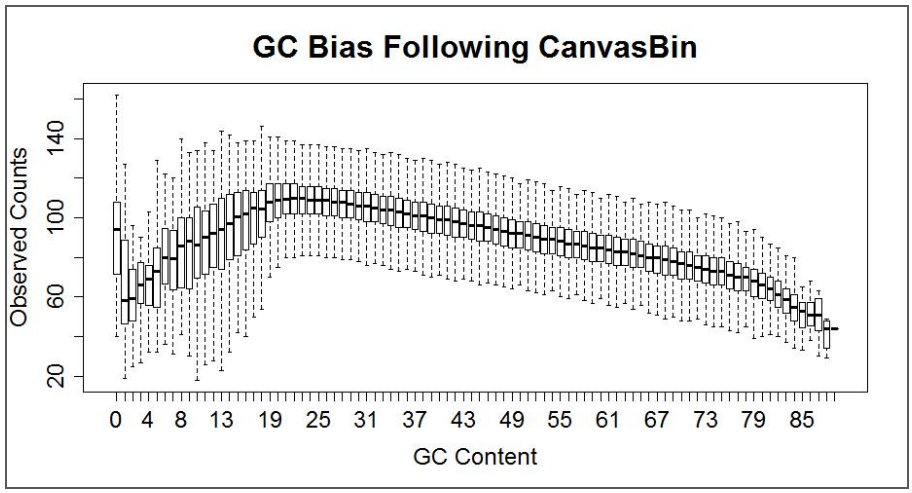
Physical Size Outlier Removal

Bins likely do not have the same physical (genomic) size. The average for whole-genome sequencing runs might be approximately 1 kb. If the bins cover repetitive regions of the genome, some bins sizes might be several megabases in size. Example regions might include centromeres and telomeres. The counts in these regions tend to be unreliable so bins with extreme physical size are removed. Specifically, the 98th percentile of observed physical sizes is calculated and bins with sizes larger than this threshold are removed.

GC Content Correction

The main variability in bins counts is GC content. An example of the bias is represented in the following figure.

Figure 3 GC Bias Example



The following correction is performed:

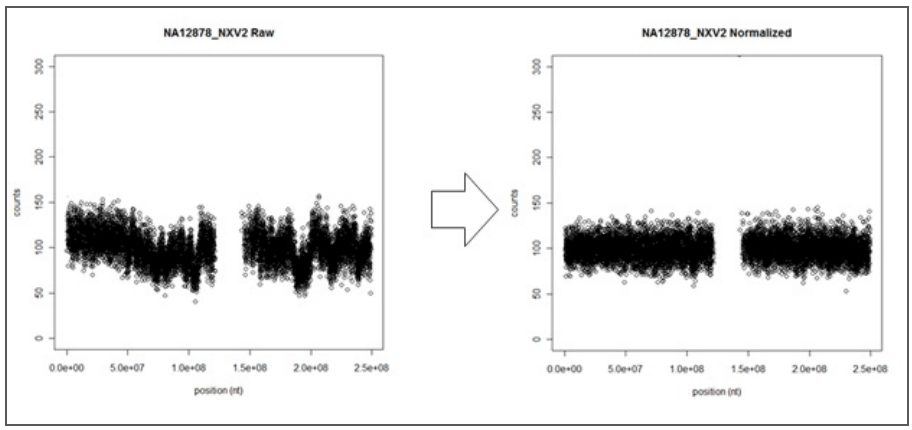
1 Bins are first aggregated according to GC content, which is rounded to the nearest integer.

2 Second, each bin count is divided by the median count of bins having the same GC content.

3 Finally, this value is multiplied by the desired average count per bin (100 by default) and rounded to the nearest integer. The effect is to flatten the midpoints of the bars in the example box-and-whisker plot.

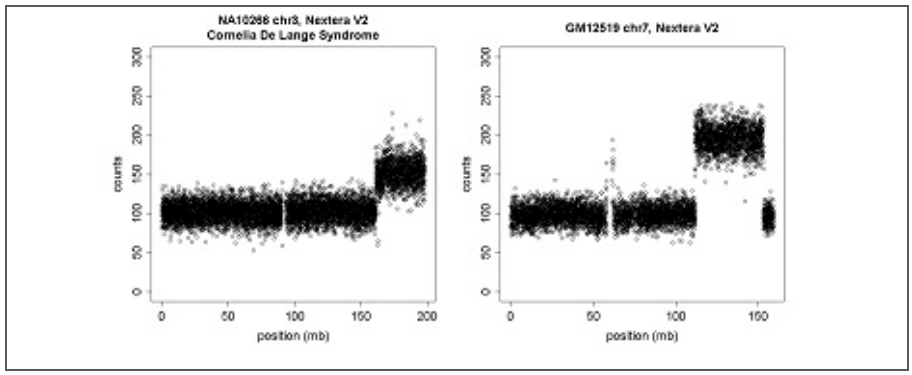
Some values for GC content have few bins so the estimate of its median is not robust. Therefore, bins are discarded when the number of bins having the same GC content is fewer than 100.

For some sample preparation schemes, GC content correction has a dramatic effect. The following figure illustrates the effect of GC content correction for a low depth sequencing experiment using the Nextera library preparation method. The figure on the left shows bins counts as a function of chromosome position before normalization. The figure on the right shows the result after GC content correction.



For whole-genome sequencing experiments, the typically median absolute deviations (MADs) are 10.3, which is close to the expected value of 10. The expected value is predicted using the Poisson model for an average count of 100 and indicates that little bias remains following GC content correction.

It is important to note that the normalization signal does not dampen signal from CNVs as shown in the following 2 figures. The figure on the left shows a chromosome known to harbor a single copy gain. The figure on the right shows chromosome known to harbor a double copy gain.



Partitioning

The Isaac CNV Caller partitioning implements an algorithm for identifying regions of the genome such that their average counts are statistically different than average counts of neighboring regions. The implementation is a port of the circular binary segmentation (CBS) algorithm.

The algorithm briefly considers each chromosome as a segment. The algorithm assesses each segment and identifies the pair of bins for which the counts in the bins between them are maximally different than the counts of the rest of the bins. The statistical significance of the maximal difference is assessed via permutation testing. If the difference is statistically significant, then the procedure is applied recursively to the 2 or 3 segments created by partitioning the current segment by the identified pair of points. Input to the algorithm is the output generated by the Isaac CNV Caller cleaning algorithm.

Because of the computational complexity of the algorithm $O(N^2)$, the problem is divided into subchromosome problems followed by merging, in practice. Heuristics are used to speed up the permutation testing.

Calling

The final module of the Isaac CNV Caller algorithm is to assign discrete copy numbers to each of the regions identified by the Isaac CNV Caller partitioner.

A Gaussian model is used as the default calling method. In this case, both the mean and standard deviation are estimated from the data for the diploid model and adjusted for the other copy number models. For example, if the mean, μ, and standard deviation, σ, are estimated to be 100 and 15 in the diploid model, then corresponding estimates in the haploid model would be $\mu/2$ and $\sigma/2$. The mean and standard deviation are estimated using the autosomal median and MAD of counts. This model is the default as it is more appropriate in cases where the spread of counts is higher than expected from the Poisson model due to unaccounted sources of variability. An example of this case is single cell sequencing experiments where whole-genome amplification is required.

Following assignment of copy number states, neighboring regions that received the same copy number call are merged into a single region.

Phred-scaled Q-scores are assigned to each region using a simple logistic function derived using array CGH data as the gold standard. The probability of a miscall is modeled as

$$p=1-(1/((1+e^{(0.5532-0.147N)})))$$

Where N is the number of bins found within the nondiploid region. This probability is converted to a Q-score by

$$q=-10 \log p$$

This estimate is likely conservative as it is derived from array CGH. Importantly, Q-scores are a function of number of bins, not genomic size, so they are applicable to experiments of any sequencing depth, including low-depth cytogenetics screening.

The coordinates of nondiploid regions and their Q-scores are output to a VCF file. Two filters are applied to PASS variants. First, a variant must have a Q-score of Q10 or greater. Second, a variant must be of size 10 kb, or greater.

Isaac Structural Variant Caller

Isaac Structural Variant (SV) Caller is a structural variant caller for short sequencing reads. It can discover structural variants of any size and score these variants using both a diploid genotype model and a somatic model (when separate tumor and normal samples are specified). Structural variant discovery and scoring incorporate both paired read fragment spanning and split read evidence.

Method Overview

Isaac SV Caller works by dividing the structural variant discovery process into 2 primary steps–scanning the genome to find SV associated regions and analysis, scoring, and output of SVs found in such regions.

1 **Build SV association graph**
In this step, the entire genome is scanned to discover evidence of possible SVs and large indels. This evidence is enumerated into a graph with edges connecting all regions of the genome that have a possible SV association. Edges can connect 2 different regions of the genome to represent evidence of a long-range association, or an edge can connect a region to itself to capture a local indel/small SV association. These associations are more general than a specific SV hypothesis, in that many SV candidates can be found on 1 edge, although typically only 1 or 2 candidates are found per edge.

2 **Analyze graph edges to find SVs**
The second step is to analyze individual graph edges or groups of highly connected edges to discover and score SVs associated with the edges. These substeps of this process include:
- Inference of SV candidates associated with the edge.
- Attempted assembly of the SVs break-ends.
- Scoring and filtration of the SV under various biological models (currently diploid germline and somatic).
- Output to VCF.

Capabilities

Isaac SV Caller can detect all structural variant types that are identifiable in the absence of copy number analysis and large scale de novo assembly. Detectable types are enumerated in this section.

For each structural variant and indel, Isaac SV Caller attempts to align the break-ends to base pair resolution and report the left-shifted break-end coordinate (per the VCF 4.1 SV reporting guidelines). Isaac SV Caller also reports any break-end microhomology sequence and inserted sequence between the break-ends. Often the assembly fails to provide a confident explanation of the data. In such cases, the variant is reported as IMPRECISE, and scored according to the paired-end read evidence alone.

The sequencing reads provided as input to Isaac SV Caller are expected to be from a paired-end sequencing assay that results in an inwards orientation between the 2 reads of each DNA fragment. Each read presents a read from the outer edge of the fragment insert inward.

Detected Variant Classes

Isaac SV Caller is able to detect all variation classes that can be explained as novel DNA adjacencies in the genome. Simple insertion/deletion events can be detected down to a configurable minimum size cutoff (defaulting to 51). All DNA adjacencies are classified into the following categories based on the break-end pattern:

- Deletions
- Insertions
- Inversions
- Tandem Duplications
- Interchromosomal Translocations

Known Limitations

Isaac SV Caller cannot detect the following variant types:

- Nontandem repeats/amplifications

- Large insertions—The maximum detectable size corresponds to approximately the read-pair fragment size, but note that detection power falls off to impractical levels well before this size.
- Small inversions—The limiting size is not tested, but in theory detection falls off below ~200 bases. So-called microinversions might be detected indirectly as combined insertion/deletion variants.

More general repeat-based limitations exist for all variant types:

- Power to assemble variants to break-end resolution falls to 0 as break-end repeat length approaches the read size.
- Power to detect any break-end falls to (nearly) 0 as the break-end repeat length approaches the fragment size.
- The method cannot detect nontandem repeats.

While Isaac SV Caller classifies novel DNA-adjacencies, it does not infer the higher level constructs implied by the classification. For instance, a variant marked as a deletion by Isaac SV Caller indicates an intrachromosomal translocation with a deletion-like break-end pattern. However, there is no test of depth, b-allele frequency, or intersecting adjacencies to infer the SV type directly.

Technical Assistance

For technical assistance, contact Illumina Technical Support.

Table 6 Illumina General Contact Information

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

Table 7 Illumina Customer Support Telephone Numbers

Region	Contact Number	Region	Contact Number
North America	1.800.809.4566	Italy	800.874909
Australia	1.800.775.688	Netherlands	0800.0223859
Austria	0800.296575	New Zealand	0800.451.650
Belgium	0800.81102	Norway	800.16836
Denmark	80882346	Spain	900.812168
Finland	0800.918363	Sweden	020790181
France	0800.911850	Switzerland	0800.563118
Germany	0800.180.8994	United Kingdom	0800.917.0041
Ireland	1.800.812949	Other countries	+44.1799.534000

Safety Data Sheets

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

Product Documentation

Product documentation in PDF is available for download from the Illumina website. Go to support.illumina.com, select a product, then click **Documentation & Literature**.

15050954 Rev. C

Illumina
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