illumina®

Whole Genome Sequencing Services Guide

FOR RESEARCH USE ONLY

ILLUMINA PROPRIETARY
Document # 15040892 v01
December 2015

Read Before Using this Product

This Product, and its use and disposition, is subject to the following terms and conditions. If Purchaser does not agree to these terms and conditions then Purchaser is not authorized by Illumina to use this Product and Purchaser must not use this Product.

1 **Definitions**. "**Application Specific IP**" means Illumina owned or controlled intellectual property rights that pertain to this Product (and use thereof) only with regard to specific field(s) or specific application(s). Application Specific IP excludes all Illumina owned or controlled intellectual property that cover aspects or features of this Product (or use thereof) that are common to this Product in all possible applications and all possible fields of use (the "**Core IP**"). Application Specific IP and Core IP are separate, non-overlapping, subsets of all Illumina owned or controlled intellectual property. By way of non-limiting example, Illumina intellectual property rights for specific diagnostic methods, for specific forensic methods, or for specific nucleic acid biomarkers, sequences, or combinations of biomarkers or sequences are examples of Application Specific IP. "**Consumable(s)**" means Illumina branded reagents and consumable items that are intended by Illumina for use with, and are to be consumed through the use of, Hardware. "**Documentation**" means Illumina's user manual for this Product, including without limitation, package inserts, and any other documentation that accompany this Product or that are referenced by the Product or in the packaging for the Product in effect on the date of shipment from Illumina. Documentation includes this document. "**Hardware**" means Illumina branded instruments, accessories or peripherals. "**Illumina**" means Illumina, Inc. or an Illumina affiliate, as applicable. "**Product**" means the product that this document accompanies (e.g., Hardware, Consumables, or Software). "**Purchaser**" is the person or entity that rightfully and legally acquires this Product from Illumina or an Illumina authorized dealer. "**Software**" means Illumina branded software (e.g., Hardware operating software, data analysis software). All Software is licensed and not sold and may be subject to additional terms found in the Software's end user license agreement. "**Specifications**" means Illumina's written specifications for this Product in effect on the date that the Product ships from Illumina.

2 **Research Use Only Rights**. Subject to these terms and conditions and unless otherwise agreed upon in writing by an officer of Illumina, Purchaser is granted only a non-exclusive, non-transferable, personal, non-sublicensable right under Illumina's Core IP, in existence on the date that this Product ships from Illumina, solely to use this Product in Purchaser's facility for Purchaser's internal research purposes (which includes research services provided to third parties) and solely in accordance with this Product's Documentation, **but specifically excluding any use that** (a) would require rights or a license from Illumina to Application Specific IP, (b) is a re-use of a previously used Consumable, (c) is the disassembling, reverse-engineering, reverse-compiling, or reverse-assembling of this Product, (d) is the separation, extraction, or isolation of components of this Product or other unauthorized analysis of this Product, (e) gains access to or determines the methods of operation of this Product, (f) is the use of non-Illumina reagent/consumables with Illumina's Hardware (does not apply if the Specifications or Documentation state otherwise), or (g) is the transfer to a third-party of, or sub-licensing of, Software or any third-party software. All Software, whether provided separately, installed on, or embedded in a Product, is licensed to Purchaser and not sold. Except as expressly stated in this Section, no right or license under any of Illumina's intellectual property rights is or are granted expressly, by implication, or by estoppel.

Purchaser is solely responsible for determining whether Purchaser has all intellectual property rights that are necessary for Purchaser's intended uses of this Product, including without limitation, any rights from third parties or rights to Application Specific IP. Illumina makes no guarantee or warranty that purchaser's

specific intended uses will not infringe the intellectual property rights of a third party or Application Specific IP.

3 **Regulatory**. This Product has not been approved, cleared, or licensed by the United States Food and Drug Administration or any other regulatory entity whether foreign or domestic for any specific intended use, whether research, commercial, diagnostic, or otherwise. This Product is labeled For Research Use Only. Purchaser must ensure it has any regulatory approvals that are necessary for Purchaser's intended uses of this Product.

4 **Unauthorized Uses**. Purchaser agrees: (a) to use each Consumable only one time, and (b) to use only Illumina consumables/reagents with Illumina Hardware. The limitations in (a)-(b) do not apply if the Documentation or Specifications for this Product state otherwise. Purchaser agrees not to, nor authorize any third party to, engage in any of the following activities: (i) disassemble, reverse-engineer, reverse-compile, or reverse-assemble the Product, (ii) separate, extract, or isolate components of this Product or subject this Product or components thereof to any analysis not expressly authorized in this Product's Documentation, (iii) gain access to or attempt to determine the methods of operation of this Product, or (iv) transfer to a third-party, or grant a sublicense, to any Software or any third-party software. Purchaser further agrees that the contents of and methods of operation of this Product are proprietary to Illumina and this Product contains or embodies trade secrets of Illumina. The conditions and restrictions found in these terms and conditions are bargained for conditions of sale and therefore control the sale of and use of this Product by Purchaser.

5 **Limited Liability. TO THE EXTENT PERMITTED BY LAW, IN NO EVENT SHALL ILLUMINA OR ITS SUPPLIERS BE LIABLE TO PURCHASER OR ANY THIRD PARTY FOR COSTS OF PROCUREMENT OF SUBSTITUTE PRODUCTS OR SERVICES, LOST PROFITS, DATA OR BUSINESS, OR FOR ANY INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY, CONSEQUENTIAL, OR PUNITIVE DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH, WITHOUT LIMITATION, THE SALE OF THIS PRODUCT, ITS USE, ILLUMINA'S PERFORMANCE HEREUNDER OR ANY OF THESE TERMS AND CONDITIONS, HOWEVER ARISING OR CAUSED AND ON ANY THEORY OF LIABILITY (WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE).**

6 **ILLUMINA'S TOTAL AND CUMULATIVE LIABILITY TO PURCHASER OR ANY THIRD PARTY ARISING OUT OF OR IN CONNECTION WITH THESE TERMS AND CONDITIONS, INCLUDING WITHOUT LIMITATION, THIS PRODUCT (INCLUDING USE THEREOF) AND ILLUMINA'S PERFORMANCE HEREUNDER, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, SHALL IN NO EVENT EXCEED THE AMOUNT PAID TO ILLUMINA FOR THIS PRODUCT.**

7 **Limitations on Illumina Provided Warranties. TO THE EXTENT PERMITTED BY LAW AND SUBJECT TO THE EXPRESS PRODUCT WARRANTY MADE HEREIN ILLUMINA MAKES NO (AND EXPRESSLY DISCLAIMS ALL) WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THIS PRODUCT, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR ARISING FROM COURSE OF PERFORMANCE, DEALING, USAGE OR TRADE. WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, ILLUMINA MAKES NO CLAIM, REPRESENTATION, OR WARRANTY OF ANY KIND AS TO THE UTILITY OF THIS PRODUCT FOR PURCHASER'S INTENDED USES.**

8 **Product Warranty**. All warranties are personal to the Purchaser and may not be transferred or assigned to a third-party, including an affiliate of Purchaser. All warranties are facility specific and do not transfer if the Product is moved to another facility of Purchaser, unless Illumina conducts such move.

- a **Warranty for Consumables**. Illumina warrants that Consumables, other than custom Consumables, will conform to their Specifications until the later of (i) 3 months from the date of shipment from Illumina, and (ii) any expiration date or the end of the shelf-life pre-printed on such Consumable by Illumina, but in no event later than 12 months from the date of shipment. With respect to custom Consumables (i.e., Consumables made to specifications or designs made by Purchaser or provided to Illumina by, or on behalf of, Purchaser), Illumina only warrants that the custom Consumables will be made and tested in accordance with Illumina's standard manufacturing and quality control processes. Illumina makes no warranty that custom Consumables will work as intended by Purchaser or for Purchaser's intended uses.
- b **Warranty for Hardware**. Illumina warrants that Hardware, other than Upgraded Components, will conform to its Specifications for a period of 12 months after its shipment date from Illumina unless the Hardware includes Illumina provided installation in which case the warranty period begins on the date of installation or 30 days after the date it was delivered, whichever occurs first ("Base Hardware Warranty"). "Upgraded Components" means Illumina provided components, modifications, or enhancements to Hardware that was previously acquired by Purchaser. Illumina warrants that Upgraded Components will conform to their Specifications for a period of 90 days from the date the Upgraded Components are installed. Upgraded Components do not extend the warranty for the Hardware unless the upgrade was conducted by Illumina at Illumina's facilities in which case the upgraded Hardware shipped to Purchaser comes with a Base Hardware Warranty.
- c **Exclusions from Warranty Coverage**. The foregoing warranties do not apply to the extent a non-conformance is due to (i) abuse, misuse, neglect, negligence, accident, improper storage, or use contrary to the Documentation or Specifications, (ii) improper handling, installation, maintenance, or repair (other than if performed by Illumina's personnel), (iii) unauthorized alterations, (iv) Force Majeure events, or (v) use

with a third party's good not provided by Illumina (unless the Product's Documentation or Specifications expressly state such third party's good is for use with the Product).

d **Procedure for Warranty Coverage**. In order to be eligible for repair or replacement under this warranty Purchaser must (i) promptly contact Illumina's support department to report the non-conformance, (ii) cooperate with Illumina in confirming or diagnosing the non-conformance, and (iii) return this Product, transportation charges prepaid to Illumina following Illumina's instructions or, if agreed by Illumina and Purchaser, grant Illumina's authorized repair personnel access to this Product in order to confirm the non-conformance and make repairs.

e **Sole Remedy under Warranty**. Illumina will, at its option, repair or replace non-conforming Product that it confirms is covered by this warranty. Repaired or replaced Consumables come with a 30-day warranty. Hardware may be repaired or replaced with functionally equivalent, reconditioned, or new Hardware or components (if only a component of Hardware is non-conforming). If the Hardware is replaced in its entirety, the warranty period for the replacement is 90 days from the date of shipment or the remaining period on the original Hardware warranty, whichever is shorter. If only a component is being repaired or replaced, the warranty period for such component is 90 days from the date of shipment or the remaining period on the original Hardware warranty, whichever ends later. The preceding states Purchaser's sole remedy and Illumina's sole obligations under the warranty provided hereunder.

f **Third-Party Goods and Warranty**. Illumina has no warranty obligations with respect to any goods originating from a third party and supplied to Purchaser hereunder. Third-party goods are those that are labeled or branded with a third-party's name. The warranty for third-party goods, if any, is provided by the original manufacturer. Upon written request Illumina will attempt to pass through any such warranty to Purchaser.

9 **Indemnification**.

a **Infringement Indemnification by Illumina**. Subject to these terms and conditions, including without limitation, the Exclusions to Illumina's Indemnification Obligations (Section 9(b) below), the Conditions to Indemnification Obligations (Section 9(d) below), Illumina shall (i) defend, indemnify and hold harmless Purchaser against any third-party claim or action alleging that this Product when used for research use purposes, in accordance with these terms and conditions, and in accordance with this Product's Documentation and Specifications infringes the valid and enforceable intellectual property rights of a third party, and (ii) pay all settlements entered into, and all final judgments and costs (including reasonable attorneys' fees) awarded against Purchaser in connection with such infringement claim. If this Product or any part thereof, becomes, or in Illumina's opinion may become, the subject of an infringement claim, Illumina shall have the right, at its option, to (A) procure for Purchaser the right to continue using this Product, (B) modify or replace this Product with a substantially equivalent non-infringing substitute, or (C) require the return of this Product and terminate the rights, license, and any other permissions provided to Purchaser with respect this Product and refund to Purchaser the depreciated value (as shown in Purchaser's official records) of the returned Product at the time of such return; provided that, no refund will be given for used-up or expired Consumables. This Section states the entire liability of Illumina for any infringement of third party intellectual property rights.

b **Exclusions to Illumina Indemnification Obligations**. Illumina has no obligation to defend, indemnify or hold harmless Purchaser for any Illumina Infringement Claim to the extent such infringement arises from: (i) the use of this Product in any manner or for any purpose outside the scope of research use purposes, (ii) the use of this Product in any manner not in accordance with its Specifications, its Documentation, the rights expressly granted to Purchaser hereunder, or any breach by Purchaser of these terms and conditions, (iii) the use of this Product in combination with any other products, materials, or services not supplied by Illumina, (iv) the use of this Product to perform any assay or other process not supplied by Illumina, or (v) Illumina's compliance with specifications or instructions for this Product furnished by, or on behalf of, Purchaser (each of (i) – (v), is referred to as an "Excluded Claim").

c **Indemnification by Purchaser**. Purchaser shall defend, indemnify and hold harmless Illumina, its affiliates, their non-affiliate collaborators and development partners that contributed to the development of this Product, and their respective officers, directors, representatives and employees against any claims, liabilities, damages, fines, penalties, causes of action, and losses of any and every kind, including without limitation, personal injury or death claims, and infringement of a third party's intellectual property rights, resulting from, relating to, or arising out of (i) Purchaser's breach of any of these terms and conditions, (ii) Purchaser's use of this Product outside of the scope of research use purposes, (iii) any use of this Product not in accordance with this Product's Specifications or Documentation, or (iv) any Excluded Claim.

d **Conditions to Indemnification Obligations**. The parties' indemnification obligations are conditioned upon the party seeking indemnification (i) promptly notifying the other party in writing of such claim or action, (ii) giving the other party exclusive control and authority over the defense and settlement of such claim or action, (iii) not admitting infringement of any intellectual property right without prior written consent of the other party, (iv) not entering into any settlement or compromise of any such claim or action without the other party's prior written consent, and (v) providing reasonable assistance to the other party in the defense of the claim or action; provided that, the party reimburses the indemnified party for its reasonable out-of-pocket expenses incurred in providing such assistance.

e **Third-Party Goods and Indemnification**. Illumina has no indemnification obligations with respect to any goods originating from a third party and supplied to Purchaser. Third-party goods are those that are

labeled or branded with a third-party's name. Purchaser's indemnification rights, if any, with respect to third party goods shall be pursuant to the original manufacturer's or licensor's indemnity. Upon written request Illumina will attempt to pass through such indemnity, if any, to Purchaser.

Revision History

Document	Date	Description of Change
Document # 15040892 v01	December 2015	• Revised documentation reflect changes in version 6 of the Illumina FastTrack WGS pipeline. • Renamed Isaac Structural Variant Caller, Isaac CNV Caller, and Isaac Variant Caller to Manta, Canvas, and Starling, respectively.
Part # 15040892 Rev. D	June 2015	• Revised documentation to reflect changes in version 4 of the Illumina FastTrack WGS pipeline. • Renamed Manta and Canvas to Isaac Structural Variant Caller and Isaac Copy Number Variant Caller, respectively.
Part # 15040892 Rev. C	July 2014	Revised documentation to reflect changes in version 3 of the Illumina FastTrack WGS pipeline.
Part # 15040892 Rev. B	July 2013	Added Circos plot legend plus minor modifications.
Part # 15040892 Rev. A	April 2013	Initial Release.

Table of Contents

Getting Started

Chapter 1

Introduction

The Whole Genome Sequencing Service leverages a suite of proven algorithms to detect genomic variants comprehensively and accurately.

The Whole Genome Sequencing Service pipeline performs the following steps:

- Isaac aligns, trims, and flags duplicates in the raw sequence.
- Canvas generates copy number (CNV) and loss of heterozygosity (LOH) analysis calls.
- Manta generates structural variant (SV) analysis calls.
- Starling generates small variant (SNV and small indels ≤ 50 bp) analysis calls.

The variants are then annotated and the resulting statistics are compiled into a summary PDF. The callers share output, and therefore there is no double-counting of calls.

Software Packages

The Whole Genome Sequencing Service pipeline uses the following software packages. For the software versions used, see the Software Versions table in the summary PDF report included with each deliverable.

Software	Purpose
Isis (Analysis Software)	Illumina Sequence Integration Software. Framework internally utilized to run the alignment, calling, annotation, and metrics.
SAMtools	Public toolkit for working with the SAM/BAM format.
Isaac (Aligner)	Aligns reads to the reference and marks duplicates.
Starling (Small Variant Caller)	Germline SNV and indel caller.
Manta (Structural Variant Caller)	Germline and somatic structural variant caller.
Canvas (CNV Caller)	Germline and somatic copy number variant caller.
Pluggable Universal Metrics Analyzer (PUMA)	Internal use only. Framework for producing metrics from BAM and VCF File.
PUMA Metrics	Internal use only. Specific version of modules for PUMA.

Most versions of the Illumina callers are open source and available puclicly. See the Illumina GitHub for the current releases.

Data Delivery

Illumina FastTrack Services currently provides data delivery through the following choices.

Illumina Hard Drive Data Delivery

Illumina FTS ships data on 1 or more hard drives. The hard drives are formatted with the NTFS file system and can optionally be encrypted.

The data on the hard drive are organized in a folder structure with 1 top-level folder per sample or analysis.

Illumina Cloud Data Delivery

Illumina FTS uploads data to a cloud container. Illumina currently supports uploads to the Amazon S3 service. Upload data are organized per upload batch by date with an Illumina_FTS prefix. For example, a sample in a batch uploaded on February 1, 2015 would be found in the container with the prefix Illumina_FTS/20140201/SAMPLE_BARCODE. Contact your FastTrack Services project manager to enable cloud delivery.

Analysis Deliverables

Chapter 2

Overview

This section details the files and folder structure for the Whole Genome 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.

Result Folder Structure

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

- **[Sample_Barcode]**
  - **[Sample_Barcode].SummaryReport.csv**—Summary report in *.csv format
  - **[Sample_Barcode].SummaryReport.pdf**—Summary report in *.pdf format
  - **Assembly**
    - **[Sample_Barcode].bam**—Archival *.bam file for sample
    - **[Sample_Barcode].bam.bai**—Index for *.bam file
  - **Genotyping**
    - **[Sample_Barcode]_idats**—Folder containing genotyping intensity data files for the sample (*.idat files) and genotyping sample sheet
    - **[Sample_Barcode].Genotyping.vcf.gz**—Genotyping SNPs mapped to reference in *.vcf format. This VCF format contains unmapped SNPs and is not fully compatible with the VCF specification.
    - **[Sample_Barcode].GenotypingReport.txt**—Genotyping SNPs tab delimited report output from GenomeStudio.
  - **Variations**
    - **[Sample_Barcode].vcf.gz**—Single nucleotyide polymorphism (SNVs) and small insertion/deletion (1 bp–50 bp) calls in *.vcf format.
    - **[Sample_Barcode].SV.vcf.gz**—Large Structural Variation calls (51 bp–10 kb) and copy number calls (10 kb+) in *.vcf format.
    - **[Sample_Barcode].genome.vcf.gz**—Genome *.vcf file containing SNVs, indels, and reference covered regions
- md5sum.txt—Checksum file for confirming file consistency.

NOTE

All the VCF files that Illumina provides are compressed and indexed using tabix. For details about tabix, see the tabix manual in SAMtools (at samtools.sourceforge.net/tabix.shtml).

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

The assembly folder contains the sequence data used to assemble the sample genome.

BAM File

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. To reduce the data storage footprint without compromising accuracy, Illumina has reduced quality score resolution in BAM files. The more commonly used 40+ possible Q-scores have been reduced to 8 bins.

For details about the reduced storage requirements, see the Reducing Whole-Genome Data Storage Footprint white paper on the Illumina product literature page.

BAM Index

This file is 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 1 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:C0L0AACXX:1:none
@RG ID:1 PL:ILLUMINA SM:NA12878 PU:C0L54ACXX:7:none
@RG ID:2 PL:ILLUMINA SM:NA12878 PU:C0L54ACXX:8:none
```

In the example, the read group 0 is derived from the flow cell barcode ID C0L0AACXX, lane 1, without a specified index for sample NA12891. In this example, read groups 1 and 2 are from a different flow cell C0L54ACXX, lanes 7 and 8.

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
   ":"cluster-id-alt"
```

ID	Description
cluster-id	Unpadded 0-based cluster id in the order in which the clusters appear within the tile.
flowcell-id	Flow cell barcode.
cluster- id-alt	In cases where the x:y coordinates from the flow cell were preserved, this column contains the y-coordinate, whereas the cluster-id contains the x-coordinate. Otherwise this column always contains "0".
lane-number	Lane number 1–8.
tile-number	Unpadded tile number.

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.	Isaac does not produce secondary alignments.
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.

Field	Description
AS	Pair alignment score.
BC	Barcode string.
NM	Edit distance (mismatches and gaps) including the soft-clipped parts of the read.

Field	Description
OC	Original CIGAR for the realigned reads.
RG	Isaac read groups correspond to unique flow cell-lane-barcodes.
SM	Single read alignment score.

Mapping Quality (MAPQ)

For pairs that match the dominant template orientation, the MAPQ value in the AS field is capped. 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

If available, variants called using the Infinium platform are compared to sequencing calls to confirm identity and make sure that data are of high quality. This folder contains the results of the genotyping SNP calls and the necessary files needed to regenerate them.

For software download files and documentation, see the GenomeStudio support page on the Illumina website.

Genotyping Intensity Data Files Folder

The [Sample_Barcode]_idats folder is contains the GRN.idat and RED.idat intensity files and the sample sheet for a genotyping sample.

These files along with the manifest, cluster, and genotyping product files can be imported into the Illumina GenomeStudio software genotyping module to reproduce the genotyping calls. The product files are on the downloads tab of the array support page.

To find the version of array chip used for your project, refer to the sample sheet in each sample intensity data files folder. If available, the *.gtc files are also included.

Genotyping VCF File

The [Sample_Barcode].Genotyping.vcf.gz file contains the genotyping SNPs in VCF format. The genotyping SNPs were mapped to the reference using megaBLAST and filtered for unique mappings.

The following filters are applied to the variants:

- Intensity only SNPs
- Any match not aligning to the SNP
- Any probe with a hamming distance greater than or equal to 5
- Any probe where the highest scoring mapping site is not the best matching site (ie, there is another site or sites within an identical hamming distance)

Any genotyping probe not matching the reference or excluded from the mapping is mapped to chromosome “NA” in the VCF file.

NOTE
Because of the additions to the Genotyping VCF file to account for unmapped probes, the files are not completely compatible with standard VCF specifications. However, you can still use most tools designed to work with VCF files.

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

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.
NOCALL	Genotype value was not called on array.

Genotyping Report

The [Sample_Barcode].GenotypingReport.txt file contains the genotyping report that is output from the GenomeStudio Genotyping Module. Illumina provides the genotyping report as 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.
GC Score	The GenCall score. This score is a quality metric assigned to every genotype called, and generally indicates their 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 that Illumina provides is compressed and includes an index for fast, range-based access. All VCF files are annotated with the FastTrack Services Annotation Pipeline and contain additional INFO fields pertaining to the annotations. For more information, see *Illumina FastTrack Services Annotation Pipeline* on page 40.

[Sample_Barcode].vcf.gz and [Sample_Barcode].genome.vcf.gz

This file contains a combined output of all single nucleotide polymorphisms and indels, respectively, called for a sample using Starling. Small indels are limited to 50 bp. Variants > 50 bp are passed to Manta and Canvas.

The VCF file contains the following INFO, FORMAT, and FILTER fields.

INFO Fields

Field	Description
BLOCK_AVG_min30p3a	Nonvariant site block. All sites in a block are constrained to be nonvariant, have the same filter value, and have all sample values in the range [x,y], where y ≤ max(x+3,(x*1.3)). All printed site block sample values are the minimum observed in the region spanned by the block.
CIGAR	The CIGAR alignment for each alternate indel allele.
END	The end position of the region described in this record.
IDREP	Number of times RU is repeated in an indel allele.
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.
DP	Filtered base call depth used for site genotyping.

ID	Description
DPF	Base calls filtered from input before site genotyping.
DPI	Read depth associated with indel, taken from the site preceding the indel.
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.

FILTER Fields

ID	Description
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.
IndelConflict	The locus is in a region with conflicting indel calls.
LowGQX	Locus GQX is < 30 or not present.
LowGQXHetSNP	Locus GQX is < 14 for het SNP.
LowGQXHomSNP	Locus GQX is < 14 for hom SNP.
LowGQXHetIns	Locus GQX is < 6 for het insertion.
LowGQXHomIns	Locus GQX is < 6 for hom insertion.
LowGQXHetAltIns	Locus GQX is < 6 for het-alt insertion.
LowGQXHetDel	Locus GQX is < 6 for het deletion.
LowGQXHomDel	Locus GQX is < 6 for hom deletion.
LowGQXHetAltDel	Locus GQX is < 6 for het-alt deletion.
PhasingConflict	Locus read evidence displays unbalanced phasing patterns.
PLOIDY_ CONFLICT	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].SV.vcf.gz

The SV file contains structural variants (51 bp–10 kb) called from the sample using Manta and Canvas.

The SV VCF file contains the following INFO, FORMAT, and FILTER fields.

INFO

ID	Description
BND_DEPTH	Read depth at local translocation break-end.
CIEND	Confidence interval around END.
CIGAR	CIGAR alignment for each alternate indel allele.
CIPOS	Confidence interval around POS.
ColocalizedCavcas	Overlapped with a 10 kb + Canvas call.
END	End position of the variant described in this record.
EVENT	ID of event associated to break-end.
HOMLEN	Length of base pair identical microhomology at event breakpoints.
HOMSEQ	Sequence of base pair identical microhomology at event breakpoints.
IMPRECISE	Imprecise structural variation.
INV3	Inversion break-ends open 3' of reported location.
INV5	Inversion break-ends open 5' of reported location.
JUNCTION_QUAL	Provides the QUAL value for only the adjacency in question.
LEFT_SVINSSEQ	Known left side of insertion for an insertion of unknown length.
MATE_BND_DEPTH	Read depth at remote translocation mate break-end.
MATEID	ID of mate break-end.
RIGHT_SVINSSEQ	Known right side of insertion for an insertion of unknown length.
SVTYPE	Type of structural variant.
SVLEN	Difference in length between REF and ALT alleles.
SVINSLEN	Length of insertion.
SVINSSEQ	Sequence of insertion.

FORMAT Fields

Field	Description
BC	Number of bins in the region.
CN	Copy number genotype for imprecise events.
GT	Genotype.
GQ	Genotype Quality.

Field	Description
PR	Spanning paired-read support for the REF and ALT alleles in the order listed.
RC	Mean counts per bin in the region.
SR	Split reads for the REF and ALT alleles in the order listed, for reads where P(allele | read) > 0.999.

FILTER Fields

Field	Description
CLT10kb	Canvas call with length < 10 kb.
MaxDepth	Sample site depth is > 3× the mean chromosome depth near 1 or both variant break-ends.
MaxMQ0Frac	For a small variant (< 1000 bases), the fraction of reads with MAPQ=0 around either break-end exceeds 0.4.
MGE10kb	Manta DEL or DUP call with length ≥ 10 kb.
MinGQ	GQ score is < 20.
NoPairSupport	For variants significantly larger than the paired read fragment size, no paired reads support the alternate allele.
Ploidy	For DEL and DUP variants, the genotypes of overlapping variants with smaller size are inconsistent with diploid expectation.
q10	Quality < 10.

[Sample_Barcode].genome.vcf.gz

The genome VCF file contains VCF formatted output for the SNVs, indels and block compressed nonvariant position output. You can use this file to compare variants and covered regions between samples quickly and in a space-efficient manner. The FILTER and INFO fields are identical to the SNV and Indel VCF file, along with the block compressed specific flags. For more information, see *gVCF (Genome VCF)* on page 24.

Summary Report

This PDF report contains an overview of the results for the samples and contains the following sections.

Section	Description
Sample Information	Contains an overview of high-level sample and sequence quality metrics: • Sample ID—Sample identifier • Total PF Reads—Total number of reads used in the analysis • Percent Q30 Bases—Number of bases with a quality score ≥ 30 out of the total number of bases. Q-binning of the BAM file does not affect this metric.
Read level Statistics	For each of the paired-end reads, the following metrics are reported: • Total Aligned Reads—Total number of reads mapping to a chromosome • Percent Aligned Reads—Percent of reads mapping to a chromosome
Base Level Statistics	For each of the paired-end reads, the following metrics are reported: • Percent Q30 Bases—Number of bases with a quality score ≥ 30 out of the total number of bases. Q-binning of the BAM file does not affect this metric. • Total Aligned Bases—Number of bases that aligned to the reference. • Percent Aligned Bases—Percent of bases aligned to the reference. • Mismatch Rate—The percent of bases mismatching the reference for mapped reads.
Coverage Histogram	Details the overall mean depth and displays a graph of bases covered for every non-N base in the reference genome.
Variant Statistics	Breaks down SNVs and indels into total counts in overlapping regions and 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.
Structural Variants Summary	Breaks down CNV and SV output into the classes of variants called. Their total PASS count and the number of overlapping genes are based on the annotation pipeline. For more information, see *Illumina FastTrack Services Annotation Pipeline* on page 40.
Fragment Length Summary	Details the fragment length statistics for the reads used in the analysis.
Duplicate Information	Details the percent of reads marked as duplicates. Duplicate reads are marked with Read 1 or Read 2 mapped positions overlapping with the highest quality read pair left unmarked.
Analysis Details	Details the parameters and versions used in the analysis.

Analysis Overview

Chapter 3

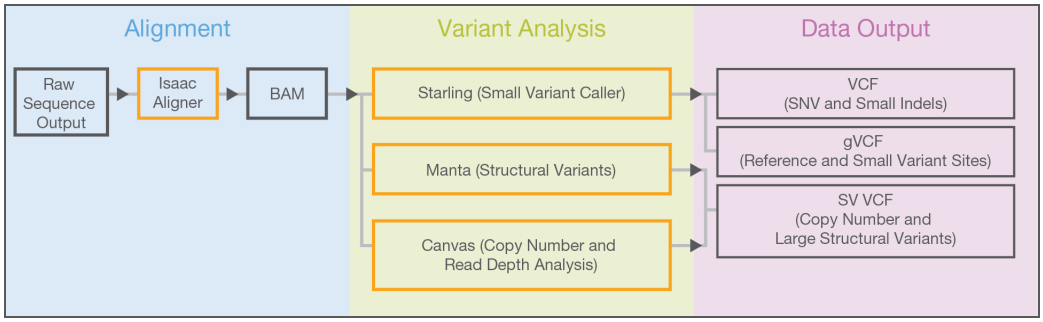
Overview

After the sequencer generates base calls and quality scores, the resulting data are first aligned to the reference genome. Then assembly and variant calling is performed.

Alignment and variant calling are performed with the Isaac Aligner, Starling, Canvas, and Manta. The following output is produced:

- 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. The reference calls are block compressed and all single nucleotide polymorphisms and indels are included. Currently Structural Variants and CNVs are kept in separate files.

Figure 1 Whole-Genome Sequencing Pipeline



Genome Specific Details

Illumina uses iGenomes as the source for the reference genomes used in alignment and assembly. The internally used genome can differ slightly from the iGenomes version in that the pseudoautosomal region (PAR) of the Y chromosome is hard masked with N's. This difference is done to avoid false mapping of reads; any mapping occurring in the PAR regions maps to the X chromosome. The reference genome used for a particular build is specified in both the BAM header and the summary output files.

The ncbi37/hg18/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

The ncbi38/hg38/GRCh38 PAR regions are defined as follows.

Name	Chr	Start	Stop
PAR#1	X	10,001	2,781,479
PAR#2	X	155,701,383	156,030,895
PAR#1	Y	10,001	2,781,479
PAR#2	Y	56,887,903	57,217,415

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.

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.

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.

Starling (Small Variant Caller)

Starling identifies single nucleotide variants (SNVs) 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 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.

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 (depth of coverage, strand bias, genotype likelihood, mapping, and base qualities, and so on). 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.

```
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 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" 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 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...

- 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 Starling 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. Starling 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: (1) describes filtering based on genotype quality; (2) describes all other filters.

NOTE
These filters are default values used in the current Starling 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.

For more information, see the small variants and genome VCF *FILTER Fields* on page 13.

Canvas (Copy Number Variations Caller)

Canvas is an algorithm for calling copy number variants from a diploid sample. Most of a normal DNA sample is diploid, or having 2 copies. Canvas identifies regions of the sample genome that are not present, or present either one time or more than 2 times in the genome. Canvas 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.

Canvas is appropriately applied to low-depth cytogenetics experiments, low-depth single-cell experiments, or whole-genome sequencing experiments. Canvas 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

Canvas 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.

Canvas 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, Canvas 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:

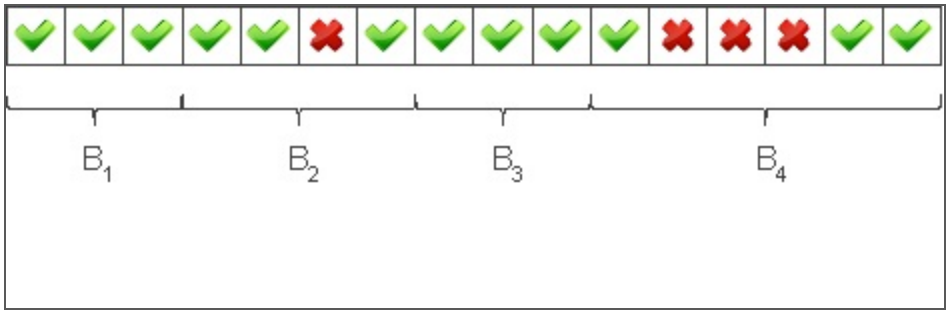
```
ATGACGATGAACGATCAGCTAAGAATACGACAATA
TGACGATGAACGATCAGCTAAGAATACGACAATAT
GAACGATCAGCTAAGAATACGACAATATCAGACAA
```

Canvas 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

Canvas 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

Canvas cleaning comprises the following 3 procedures that remove outliers and systematic biases from the count data computed in the 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 Canvas 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.

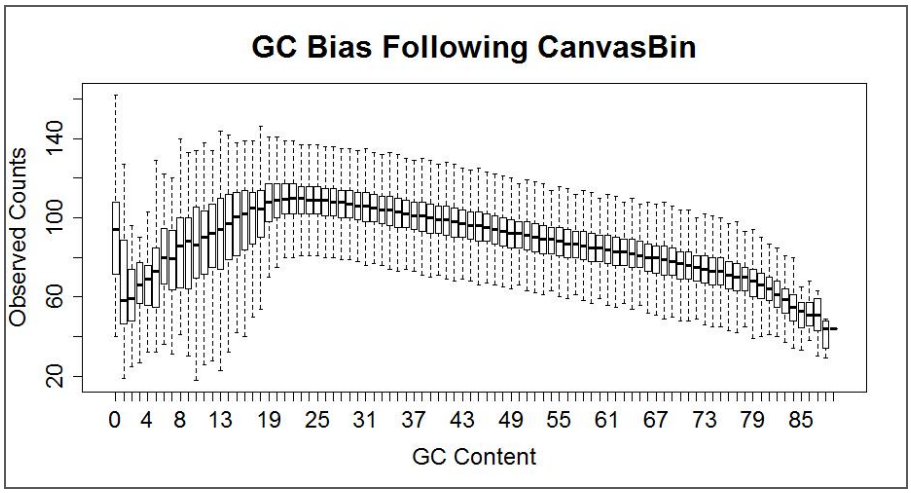
Physical Size Outlier Removal

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 2 GC Bias Example



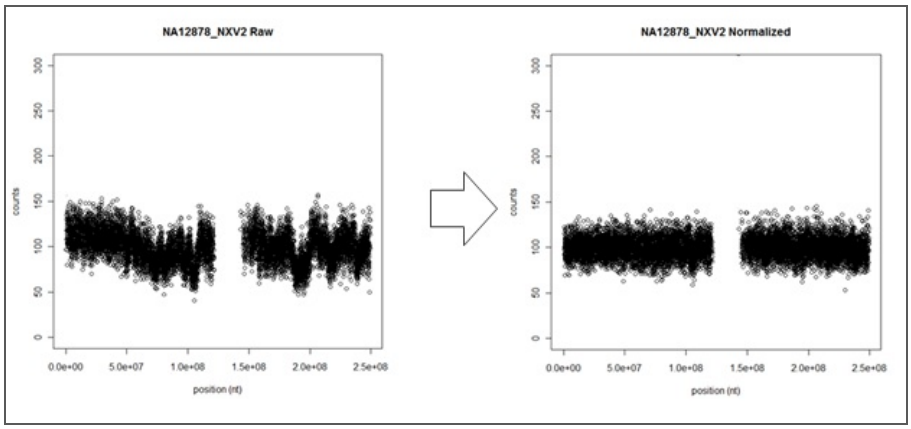
The following correction is performed:

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.

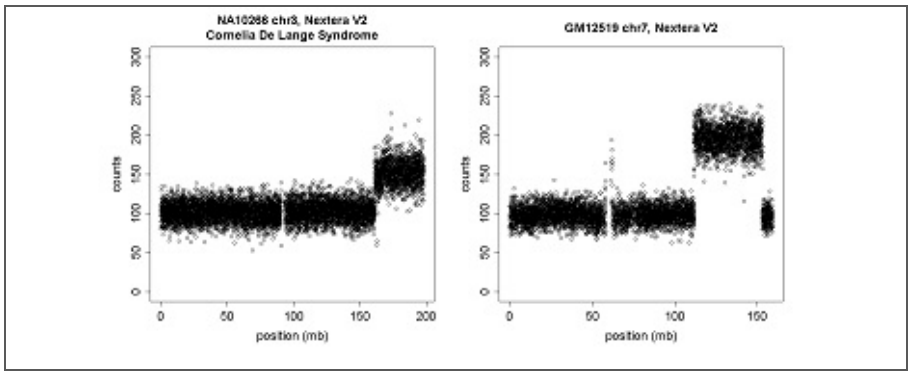
Figure 3 GC Content Correction



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.

Figure 4 Chromosomal Copy Gain



Partitioning

Canvas 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 Canvas 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 Canvas algorithm is to assign discrete copy numbers to each of the regions identified by the Canvas 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.

Manta (Large Indel and Structural Variant Caller)

The large indel and structural variant calling method (Manta) 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.

For more information, see the publication Manta: Rapid detection of structural variants and indels for clinical sequencing applications or the Manta GitHub.

Chen,X., Schulz-Trieglaff,O., Shaw,R. *et al.* (2015) Manta: Rapid detection of structural variants and indels for clinical sequencing applications. *Bioinformatics*. Advance online publication. doi: 10.1101/024232

Method Overview

Manta 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**
   Scan the entire genome 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**
   Analyze individual graph edges or groups of highly connected edges to discover and score SVs associated with the edges. The 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

Manta 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, Manta 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). Manta 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 Manta 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

Manta 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

Manta 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.

  The FastTrack Whole-Genome Sequencing service reports called variants that are 50–10 kb in 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 Manta 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 Manta 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.

Appendix

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 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 sort 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.

```
# Sort the bam file by name and output to sorted_by_name.bam
$ samtools/samtools-0.1.19/samtools sort -n -@ 4 -m 1G
   Example.bam Example_sorted

Bam Size: 79G
Wall Clock Time: 3 hrs 5 min
```

Optional Parameters:

```
-@ 4 : This option tells samtools to run 4 threads
-m 1G : This option tells samtools to use 1Gb of memory per
   thread.
```

For additional information about SAMtools, see samtools.sourceforge.net/

Reads Extraction Using SAMtools Flags

The BAM/SAM format contains a “bitwise flag” column that contains a hexadecimal, which defines the nature of that read. SAMtools allows you to easily filter on reads based

on this flag. There are 12 types of these flags. Using the including (-f) or the excluding (-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
```

Illumina FastTrack Services Annotation Pipeline

The FastTrack pipeline is an internal pipeline that provides the following annotations.

NOTE
These versions are specific to the time of publication of this document and can change with later updates. To determine the versions used, see the VCF file headers.

Source	Version	Release Date
dbSNP	144	06/06/2015
COSMIC	v73	06/06/2015
1000 Genomes Project	Phase 3 v5a	05/27/2013
EVS	V2	11/13/2013
ClinVar	Unknown	09/02/2015
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 (Gene Index)

Technical Assistance

For technical assistance, contact Illumina Technical Support.

Table 2 Illumina General Contact Information

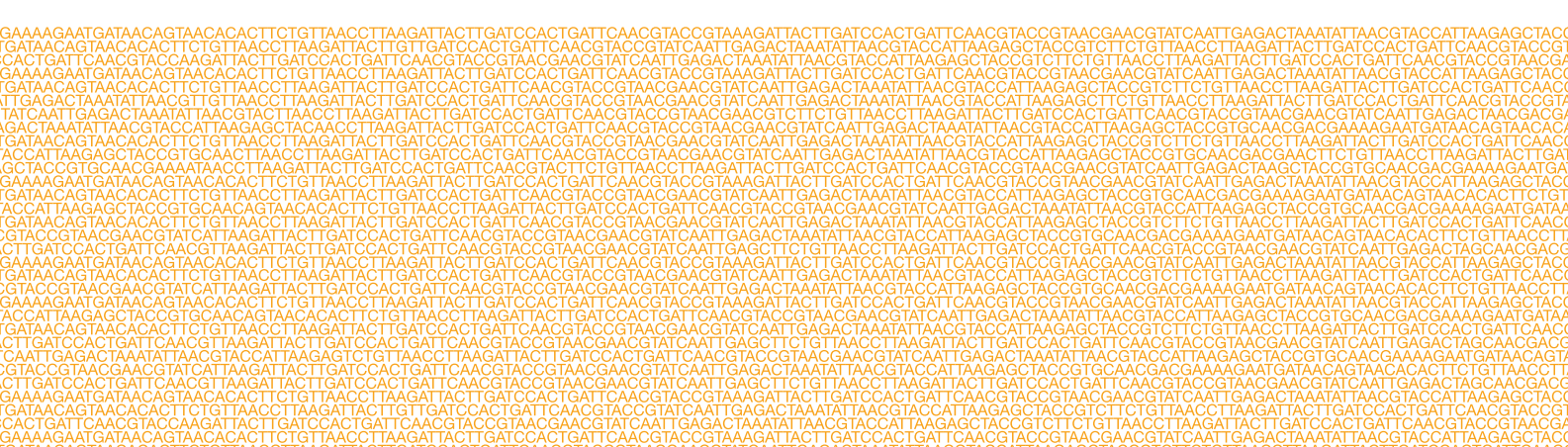
Website	www.illumina.com
Email	techsupport@illumina.com

Table 3 Illumina Customer Support Telephone Numbers

Region	Contact Number	Region	Contact Number
North America	1.800.809.4566	Japan	0800.111.5011
Australia	1.800.775.688	Netherlands	0800.0223859
Austria	0800.296575	New Zealand	0800.451.650
Belgium	0800.81102	Norway	800.16836
China	400.635.9898	Singapore	1.800.579.2745
Denmark	80882346	Spain	900.812168
Finland	0800.918363	Sweden	020790181
France	0800.911850	Switzerland	0800.563118
Germany	0800.180.8994	Taiwan	00806651752
Hong Kong	800960230	United Kingdom	0800.917.0041
Ireland	1.800.812949	Other countries	+44.1799.534000
Italy	800.874909		

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

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



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