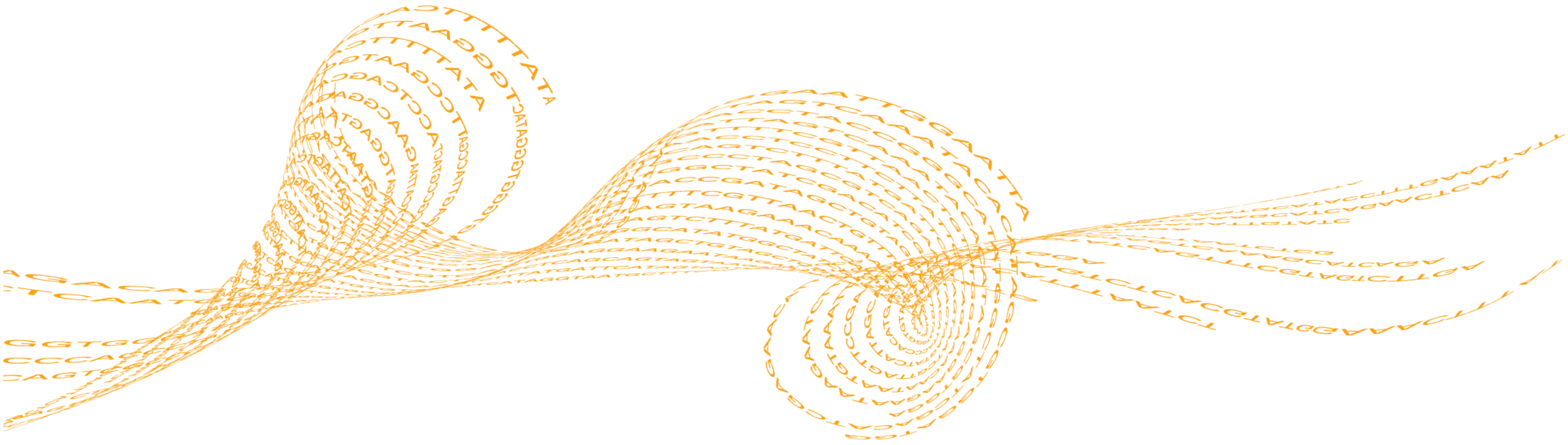
CASAVA v1.8 Changes

FOR RESEARCH USE ONLY

Introduction	2
FASTQ Files	2
FASTA Reference Genome	2
Directory Structure	3
Repeated Analysis of the Data	3
Change to the GERALD Summary File	3
Change to the Export File Format	3
Change to Quality Encoding	4
Adoption of BAM	4
Post-alignment variant-calling output	4
Appendix	5



Current as of January 5, 2011

illumina®

Introduction

Illumina has made several changes to file formats and directory structures for the upcoming release of CASAVA 1.8, which is planned for early access in late February, 2011. The changes are designed to make CASAVA more compatible with existing open-source NGS analysis software and to improve the usability of the demultiplexing workflow and the resulting directory structure. These changes, however, can impact third party tools and existing analysis pipelines at user sites that rely on the file formats and directory structures generated by previous versions of CASAVA. We are making this document available before the release of CASAVA 1.8 so that users have advance notice to make the necessary changes to existing pipelines.

The key changes are:

- The bcl converter will be distributed with CASAVA.
- The converter will produce compressed FASTQ files rather than qseq files.
- The FASTQ quality score encoding will use the standard offset value of 33 rather than the previous Illumina-specific offset value of 64.
- If samples have been multiplexed in a sequencing run using indexing, the converter will also perform demultiplexing.
- The output files will be in a directory structure organized by project and sample rather than lane and tile.
- The GERALD summary file will be modified in accordance with the new directory structure.
- The sequence output of post-alignment analysis will be a set of BAM files.

FASTQ Files

RTA will continue to produce bcl files which contain the base call and quality score information. The bcl converter will convert the bcl files to compressed FASTQ. The specifications for the FASTQ file are provided here. The FASTQ file is gzip compressed and the format has been chosen for compatibility with common aligners. FASTQ is also the format now expected by ELANDv2e, the aligner used in CASAVA 1.8. The Illumina-specific qseq format will no longer be generated, though conversion from qseq to FASTQ will be possible.

FASTA Reference Genome

The reference genomes for alignment will be provided in FASTA format. Genome squashing will occur automatically, so the squashed files will no longer be needed. The FASTA genomes for several common organisms are available through iGenomes located at: https://icom.illumina.com/Message/iGenome/.

Directory Structure

The RUN folder will contain subfolders called "Unaligned" and "Aligned". A "Build" folder can be created during the variant calling step. Additional details are provided here for the "Unaligned" folder, but the other folders have an analogous structure. The output results will be organized by project as specified by the user in the sample sheet. The project folders allow users to naturally group samples processed on a flow cell. Within the projects folder, there will be a folder for each sample. Within a sample folder, the FASTQ files for the sample will be subdivided so that each file is of a convenient size for downstream processing. The name of each FASTQ file provides the sample, the index used, the lane, and whether the file contains the first or second read. The names are based on information obtained from the sample sheet. If no sample sheet is provided, a default sample sheet is assumed. The same folder structure will accommodate single sample and indexed runs. A figure with a sample directory structure is provided here.

Repeated Analysis of the Data

If data is analyzed multiple times, the default behavior will be to repeat the previously described directory structure but to add a time stamp to the highest level folder names so that data is not overwritten by default. The user will have the option to specify the output location. By specifying a location, the user will be able to customize folder names, to combine data from multiple runs into a single build, or to overwrite previous analysis if desired.

Change to the GERALD Summary File

The summary.htm file will be modified in accordance with the directory structure changes. All information will be organized based on sample. The file will provide easy access to the relative abundance of each index in a lane, the performance of each sample within a lane and index, and the overall performance of a sample. Information regarding individual lane or tile quality is available as RTA output and can be accessed with the Sequence Analysis Viewer. The philosophy is that RTA is used to assess run quality at the level of the flow cell, while the GERALD summary is used to assess quality at the sample level.

Change to the Export File Format

The export files are the output of the alignment process, and they are intended to be an Illumina internal file format. The files are used as input to CASAVA's post-alignment analysis. A change has been made to capture information on reads that have been identified as manufacturer controls. This is done by labeling the chromosome field for these reads as "CONTROL". The utilization of the chromosome field already occurs for other types of unaligned reads, so this is a natural extension. The information is carried forward so that it will appear in the final BAM file. To minimize storage, the export files will be gzip compressed.

Change to Quality Encoding

The quality scores are transformed from integer to character so that a string can represent all of the quality scores within a read. In the CASAVA 1.8 release, we employ an ASCII offset of 33, which is the offset used in the Sanger FASTQ format. Illumina has moved away from an Illumina-specific offset, and adopted the Sanger transformation which is standard in the sequencing field For example, a Q30 base that was previously represented by the character "^" will now be represented by the character "?". The new transformation will be evident in the FASTQ file and the BAM file. The old transformation (ASCII offset of 64) will still be used in the export files, but export.txt is intended to be an internal file format.

Adoption of BAM

The BAM file format will be used to represent reads in CASAVA's post-alignment analysis, which sorts the reads according to chromosome position. This format is the standard in the field and it will be the input to the variant caller in CASAVA 1.8. The sorted.txt files will no longer be generated, thus dramatically reducing the size of the CASAVA build.

Post-alignment variant-calling output

In CASAVA 1.8 the SNP-caller has been changed to a probabilistic method which provides a predicted diploid genotype for every position in the genome together with SNP and genotype quality scores. Tab-delimited text files are produced to provide this information for every non-empty site in the genome and for all putative SNP sites. The output of the indel-caller is very similar to CASAVA 1.7, with only minor accommodations to enable overlapping indel calls. The sort.count files will no longer be produced, but there will be an option to produce the same information within the variant calling text file called sites.txt.gz.

Additional detail on all of the features of CASAVA 1.8 and all of the options available for running the program will be available in the CASAVA 1.8 User Manual when CASAVA is released.

Appendix

FASTQ format
The 1000 Genomes Project uses the minimal FASTQ project. Considering these files are produced by the DCC at EBI and NCBI, they represent the most popular format used by large projects where the NIH is involved. We will use the same approach. The FASTQ files will be gzip compressed to minimize storage. Many popular aligners are able to directly read compressed FASTQ files.
A sample entry is provided and explained below:

```
@EAS139:136:FC706VJ:2:5:1000:12850 1:Y:18:ATCACG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
```

- The first line is prefixed by the “@” symbol and contains the read name. These names are parsed until the first encountered whitespace. Due to this behavior, adding additional tags to the header line is not problematic for extant FASTQ parsers.
- The second line contains the sequence bases
- The third line is prefixed by a + symbol and sometimes repeats the read name. The read name is omitted in the minimal FASTQ case.
- The fourth line contains the base qualities where BQ + 33 = ASCII value shown in the base quality string

The header line is interpreted as follows:
@ <instrument-name>:<run ID>:<flowcell ID>:<lane-number>:<tile-number>:
<x-pos>: <y-pos> <read number>:<is filtered>:<control number>:<barcode sequence>

- Note the space between <ypos> and <read number>. In a paired end run, read 1 and read 2 will be in different FASTQ files, but we want them to have matching template names. The name up-to the space will also be used as the read name in the final BAM file.
- <read number> will typically be 1 or 2, but the field can support other values. (For example, certain indexing formats lead to 3 reads.)
- <is filtered> is Y if the read is filtered, N otherwise.
- <control number> is 0 when none of the control bits are on, otherwise it is an even number.
- <barcode sequence> represents the USE_BASES masked barcode sequence, empty otherwise.

Directory Structure Example

```
/RUN (or user specified location)
    /Unaligned
        /Project_DNA_A
            /Sample_NA10831
                NA10831_ATCACG_L002_R1_001.fastq.gz
                NA10831_ATCACG_L002_R1_002.fastq.gz
                NA10831_ATCACG_L002_R1_003.fastq.gz
                NA10831_ATCACG_L002_R2_001.fastq.gz
                NA10831_ATCACG_L002_R2_002.fastq.gz
                NA10831_ATCACG_L002_R2_003.fastq.gz
                NA10831_ATCACG_L003_R1_001.fastq.gz
                NA10831_ATCACG_L003_R1_002.fastq.gz
                NA10831_ATCACG_L003_R2_001.fastq.gz
                NA10831_ATCACG_L003_R2_002.fastq.gz
            /Sample_NA10859
                NA10859_CGATGT_L002_R1_001.fastq.gz
                NA10859_CGATGT_L002_R1_002.fastq.gz
                NA10859_CGATGT_L002_R2_001.fastq.gz
                NA10859_CGATGT_L002_R2_002.fastq.gz
       /Project_RNA_B
            /Sample_NA10861
                NA10861_CGATGT_L003_R1_001.fastq.gz
                NA10861_CGATGT_L003_R2_001.fastq.gz
            /Sample_NA10860
                NA10860_TTAGGC_L003_R1_001.fastq.gz
                NA10860_TTAGGC_L003_R2_001.fastq.gz
       /Project_Control
            /Sample_PhiX
                PhiX_NoIndex_L001_R1_001.fastq.gz
                PhiX_NoIndex_L001_R2_001.fastq.gz
       /Undetermined_indices
            /Sample_lane2
                lane2_Undetermined_L002_R1_001.fastq.gz
                lane2_Undetermined_L002_R2_001.fastq.gz
            /Sample_lane3
                lane3_Undetermined_L003_R1_001.fastq.gz
                lane3_Undetermined_L003_R2_001.fastq.gz
    /Aligned
       /Project_DNA_A
            /Sample_NA10831
                (export and summary files)
            /Sample_NA10859
                (export and summary files)
       /Project_RNA_B
            /Sample_NA10861
                (export and summary files)

            /Sample_NA10860
                (export and summary files)
       /Project_Control
            /Sample_PhiX
                (export and summary files)
    /Build
       /Project_DNA_A
            /Sample_NA10831
                (BAM and variant files)
            /Sample_NA10859
                (BAM and variant files)
       /Project_RNA_B
            /Sample_NA10861
                (BAM and variant files)
            /Sample_NA10860
                (BAM and variant files)
       /Project_Control
            /Sample_PhiX
                (BAM and variant files)
```