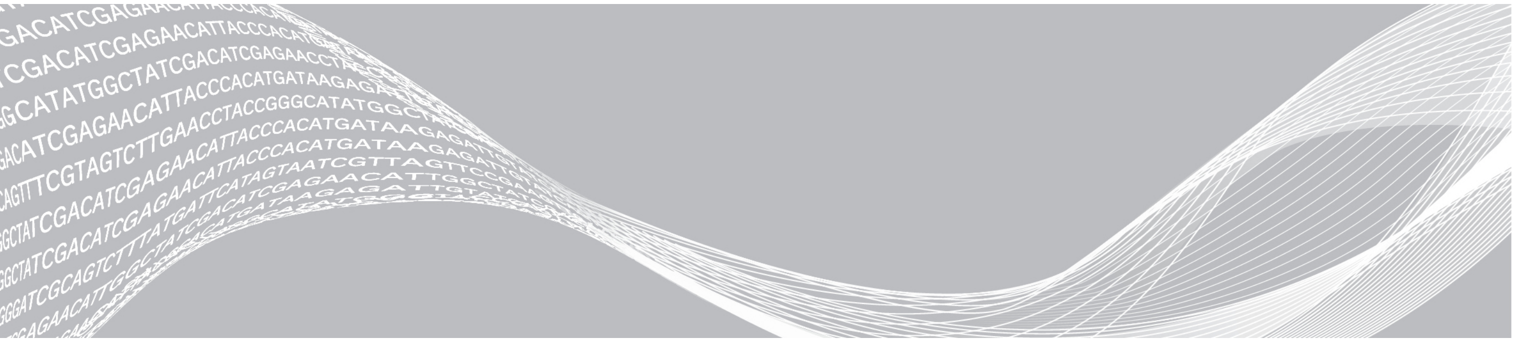
UMI Error Correction Local App

User Guide

Table of Contents



Document # 1000000037270 v00
April 2018

ILLUMINA PROPRIETARY

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

Introduction

The UMI Error Correction Local App is a Docker-based software package that analyzes sequencing reads from DNA libraries prepared using the TruSight® Oncology UMI Reagents. Unique Molecular Identifiers (UMIs) are short sequences of bases added at the beginning of library preparation that can be used to uniquely identify each molecule going into the reaction. The UMI Error Correction Local App uses the UMI information contained within the sequencing reads to collapse PCR duplicates originating from the same unique DNA fragment. Collapsing PCR duplicates results in accurate consensus reads without losing the signal of very low frequency sequence variations.

The UMI Error Correction Local App produces three BAM files per sample.

- BAM with aligned and noncollapsed sequencing reads
- BAM with aligned and collapsed sequencing reads
- BAM file with aligned, collapsed, and stitched sequencing reads

The UMI Error Correction Local App also produces a read collapsing metrics report and intermediate analysis files. Analysis of up to 96 samples (from one flow cell) is supported.

This document provides an overview of the UMI Error Correction Local App workflow, computing requirements, installation, analysis software, and analysis methods. For Research Use Only. Not for use in diagnostic procedures.

UMI Error Correction Local App Workflow

Workflow Requirements

- Basic knowledge of Linux and Docker is required.
- Supports assays prepared with TruSight Oncology UMI Reagents.
- Compatible with Sequencing on HiSeq 2500/4000. Compatibility with the NovaSeq 6000 was demonstrated but not extensively tested.
- Sequencing settings
  - Recommended sequencing runs are as follows.
    - 2 × 125 base pairs on the HiSeq 2500 with high output run mode
    - 2 × 151 base pairs on the HiSeq 4000 or NovaSeq 6000
  - Paired End (required)

Prepare Sample Sheet

To start secondary analysis, you must create a sample sheet prior to initiating a sequencing run.

- Two sample sheet templates are provided on the Illumina Support website, as follows.
  - For sequencers that use standard i5 primer notation
  - For sequencers using reverse complemented i5 primer notation

  For more information, see the *Indexed Sequencing Overview Guide* (15057455).
- The sample sheet templates contain the following parameters that are required for processing by the UMI Error Correction Software.
  - Read1UMILength 7
  - Read2UMILength 7

    - Read1StartFromCycle 9
    - Read2StartFromCycle 9
- If you are using a sample sheet within the run folder, the sample sheet must be named SampleSheet.csv.
- The Reads section must be defined in the sample sheet.
    - Maximum number of samples per analysis run is 96 for the UMI Error Correction Local App.
    - Data section—required columns and column headers are as follows.
        - Sample_ID
        - Sample_Name
        - Index
        - Index2

NOTE

The HiSeq 4000 uses reverse complements of the second index (i5).

- The Sample ID is used in output file names instead of the sample name. For an easier transition between the UMI Error Correction BaseSpace Sequence Hub App and the UMI Error Correction Local App (that does not require changing the file name), do not use underscores in the Sample ID and leave the Sample Name blank.
- Sample_ID naming conventions are as follows.
    - Alphanumeric characters, length fewer than 100 characters
    - Cannot contain nonalphanumeric special characters (* . " / \ [ ] : ; | = ,) or spaces
    - Prefix Sample_IDs with a keyword
    - Unique combinations of Sample_ID and Lane Number (required)

CAUTION

If the files will be transferred to BaseSpace Sequence Hub, do not use underscores (_) in the Sample_ID.

- When uploading data to BaseSpace Sequence Hub, do not use the 1+2+3+4 naming convention to specify multiple lane usages. Instead specify sample and lanes individually or leave the lane column blank to indicate all lanes.
- Any of the following conditions may cause an error message.
    - Missing sample name column header
    - Missing IEMFileVersion
    - Missing Header section header
    - Missing Settings section header
    - Missing Reads section header

Additional Resources

Visit theTruSight Oncology UMI Reagents support page on the Illumina website for documentation, training resources, and information about compatible Illumina products.

The following documents are available for download from the Illumina website.

Document	Description
HiSeq 2500 Sample Sheet Template (document # 1000000036306)	Used to start a sequencing run on the HiSeq 2500 System.
HiSeq 4000 Sample Sheet Template (document # 1000000036305)	Used to start a sequencing run on the HiSeq 4000 System. The HiSeq 4000 uses reverse complements of the second index (i5).
NovaSeq 6000 Sample Sheet Template (document # 1000000036332)	Used to start a sequencing run on the NovaSeq 6000.

Versions

The following analysis components are used in the UMI Error Correction Local App.

Software Component	Vendor	Version
Burrows-Wheeler Aligner	Open Source	0.7.15
SAMTools	Open Source	1.3.1

Resources

The resources folder contains the TruSight Tumor 170 DNA manifest file and the hg19 reference genome used in the UMI Error Correction Local App.

The TruSight Tumor 170 DNA manifest should only be used when analyzing libraries prepared with the TruSight Tumor 170 DNA kit and the TruSight Oncology UMI Reagents.

The only reference genome allowed for alignment by BWA in the UMI Error Correction Local App is the pseudoautosomal region masked UCSC version of hg19. Providing a different reference genome prevents the analysis from running.

Glossary

Term	Definition
BAM	A standard (usually compressed) file format for representing aligned reads. The file is a binary version of the Sequence Alignment Map file standard.
BCL	Base call files produced by Illumina sequencers.
bcl2fastq	An Illumina program that translates BCL files into FASTQ files and labels the resulting reads with the UMIs.
Collapse	To derive a consensus read from all the reads in a family.
Correction	To alter a UMI or positional grouping to make that grouping more accurate based on known valid UMIs and nearby families.
Duplex-Family	Two families that originate from the two strands of dsDNA are collapsed into a duplex-family.
Family	A group of read pairs that share the same position and UMI tags. Also referred to as a "Bag" in some metrics.
FASTQ	A standard file format for representing unaligned reads and their corresponding quality data.
Manifest	A set of genomic intervals (or regions).

Term	Definition
Noise Allele Frequency/ Error Rate	Average of all allele frequency values between 0 and 0.05, excluding loci overlapping with known germline mutations from 1000 genome data set.
Read Pair	A Read 1/Read 2 pair of reads from the same cluster.
ReCo	The ReadCollapser program.
Stitcher	The read stitching program.
UMI	Unique Molecular Identifier.

Computing Requirements

The following are the minimum software and hardware requirements per UMI Error Correction Local App analysis, which supports a maximum of 96 libraries (DNA only) per run.

Software

The UMI Error Correction Local App requires Linux Kernels 3.10 - 3.13 or 3.18+ and Docker Engine 17.03 or later. The software was developed and tested using CentOS 7.3. Other systems might not be supported.

Hardware

The UMI Error Correction Local App requires the following hardware.

Minimum

- 128 GB RAM
- 64-bit CPU with 12 physical cores, 24 threads
- 1 TB HDD Free Space

Recommended

- 128 GB RAM, 2400 MHz DDR4 RDIMM ECC
- Dual Intel Xeon 64-bit CPUs; E5-2620 v4 or better
- 1 TB HDD Free Space
- Hardware RAID controller (1 GB cache)
- 2x 2 TB SATA 7.2k RPM HDD (enterprise grade) in RAID 1 configuration
- Dedicated machine or node

Install the UMI Error Correction Local App

Before installing the UMI Error Correction Local App, make sure that you have completed the following prerequisite tasks.

- Verify that you have sudo privileges. If you do not, contact your system administrator.
- Install and set up Docker. For more information, see https://www.docker.com.
- Install and configure AWS Command Line Interface. For more information, see https://aws.amazon.com/cli/.

Installation

1. Obtain the Docker image, resource folder, licensing file, download commands, and checksum values from your local area Field Application Specialist.

2. Download the Docker image and resource folder using the commands provided from the support page.

3. Verify the Docker image file by calculating the md5 hash value using the following command.

   ```
   # {DOWNLOAD_PATH}: Path to location of downloaded local app docker image
   $ md5sum {DOWNLOAD_PATH}/UMI_Error_Correction_Local_App_
      {AppVersionNumber}.tgz
   ```

   The calculated md5sum must match the value provided from the download instructions.

4. Verify the resources file by calculating the md5 hash value using the following command.

   ```
   # {DOWNLOAD_PATH}: Path to location of downloaded local app docker image
   $ md5sum {DOWNLOAD_PATH}/UMI_Error_Correction_Local_App_Resources_
      {AppVersionNumber}.tgz
   ```

   The calculated md5sum must match the value provided from the downloaded md5sum.txt file.

5. Extract the resources file using the following command.

   ```
   # Command to extract downloaded resources tar file contents
   # {DOWNLOAD_PATH}: Path to location of downloaded local app docker image
   # {GENOME_RESOURCE_PATH}: Output path to location for extracted files

   $ tar -zxvf {DOWNLOAD_PATH}/UMI_Error_Correction_Local_App_Resources_
      {AppVersionNumber}.tgz -C {GENOME_RESOURCE_PATH}
   ```

6. Load the Docker image using the following command.

   ```
   # Command to load docker image
   # {DOWNLOAD_PATH}: Path to location of downloaded local app docker image

   $ sudo docker load < {DOWNLOAD_PATH}/UMI_Error_Correction_Local_App_
      {AppVersionNumber}.tgz
   ```

7. Run the following command to list docker images. Verify that the list includes an image matching the Repository, Tag, and Image ID shown in the table below.

   ```
   # Command to list docker images
   $ sudo docker images
   ```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
umi-error-correction-local-app	1.0.0.0	65d9a766999f	3 days ago	

NOTE

The instructions for running the UMI Error Correction Local App assume that you have root or sudo privileges to perform Docker run commands. System packages are required to install aws-cli onto your system and to run the UMI Error Correction Pipeline via Docker. If you do not have sudo privileges, please contact your system administrator. For more details on Docker permission requirements, please review the following guide: https://docs.docker.com/engine/installation/linux/linux-postinstall/.

Run the UMI Error Correction Local App

Docker Summary

Docker is a software container platform. A container is a way to package software that can run in an isolated environment on a shared operating system. Containers are an efficient, lightweight, self-contained system and guarantee that the software always runs the same way, regardless of where it is deployed. Unlike VMs, containers do not bundle a full operating system. Only libraries and settings required to make the software work are needed.

A brief tutorial of how to use Docker is available at https://docs.docker.com/get-started/.

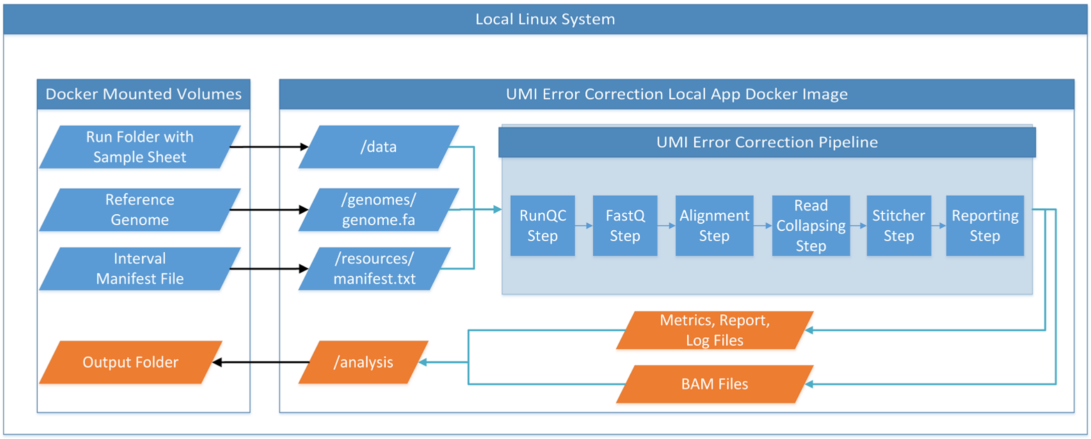
Basic UMI Error Correction Local App Docker Image Workflow

The required inputs to the UMI Error Correction Local App Docker image, shown below, are the following.

- Path to the run folder (containing the sample sheet). Do not use an original run folder for the workflow without a backup copy.
- Path to the reference genome.
- Path to manifest file.
- Path to the analysis output folder.

These paths are used to mount the volumes to share the space with the Docker image for reading and writing of the data.

Figure 1 Basic UMI Error Correction Local App Docker Image Workflow



Basic Operation

- Launch one UMI Error Correction Local App container at a time. Launching multiple containers from the same run folder causes file access problems if not skipping steps. Run the following command to check for running containers on the system before starting a new analysis:

  ```
  $ sudo docker ps
  ```

- Illumina recommends that Local system Paths specified in the -v (volume mount) command line to run the Local App be absolute paths (/home/user/path instead of ~/path).

- Identify the local system User ID to be used to run the process. The command needs the ID, not user name. Note that one of the paths is mounted in read-only mode and the others require read and write permissions. To run the processes as your user ID, get your unique system ID as shown in the example below. To run the process using a different system user ID, check with your system administrator to ascertain the proper UID.

  ```
  # check the user id of the actively logged in account:
  $ echo $UID
  1002
  ```

- Avoid using special characters (eg &, *, #, spaces) in all mount paths (special characters can cause issues with Docker).
- For each run, the following four volume mounts must be specified.
  - GENOMES_PATH—Full path to the folder containing the genome.fa file. Read-only permissions are required for the user ID running the pipeline.
  - MANIFEST_FILE—Full path to manifest file. Read-only permissions are required for the user ID running the pipeline.
  - RUN_PATH—Full path to the folder containing the data set being processed. Read / Write permissions are required for the user ID running the pipeline.
  - ANALYSIS_PATH—This folder contains the output from the analysis run. Read / Write permissions are required for the user ID running the pipeline.

Manifest File Format

The UMI Error Correction Local App accepts any manifest files that conform to the following formatting rules based on Nextera Rapid Capture manifests in .txt or .bed format.

- Required
  - Manifest chromosomes corresponding to hg19 (eg, chromosome 1 = chr1 and not chromosome 1 = 1)
  - Tab-delimited file
  - [Regions] section is required
  - Column headers
    - Name
    - Chromosome
    - Amplicon Start or Start
    - Amplicon End or End
  - Data type by column header (column, data type)
    - Name, string
    - Chromosome, string
    - Start, integer
    - End, integer
  - Unique alphanumeric names for each manifest region
  - Inclusive manifest regions
  - One based indexing for start and end positions
  - Custom Manifest regions containing single base overlaps are not supported.

- Optional
  - Column headers
    - Target Region
    - Upstream Probe Length
    - DownStream Probe Length
  - Data type by column header (column, data type)

    Target Region, string

    Upstream Probe Length, int

    Downstream Probe Length, int
  - Optional [Headers]
    - Recommended headers: Manifest File version and Reference Genome

Example TruSight Tumor 170 DNA Manifest:

```
#Experiment
#Final DNA Probes 04/28/2016
[Header]
Manifest Version Release2HDNA_manifest_160428
ReferenceGenome Homo_sapiens\UCSC\hg19\Sequence\WholeGenomeFASTA
[Regions]
Name Chromosome Amplicon Start Amplicon End Target Region Upstream Probe
   Length Downstream Probe Length
chr1_iSNP_1_1 chr1 4367257 4367353 chr1_iSNP_1 0 0
PIK3CD_Exon3_2 chr1 9770514 9770654 PIK3CD_Exon3 0 0
PIK3CD_Exon4_3 chr1 9775599 9775827 PIK3CD_Exon4 0 0
```

Example Scenario—To perform a standard end to end run with all steps use the following default entry point.

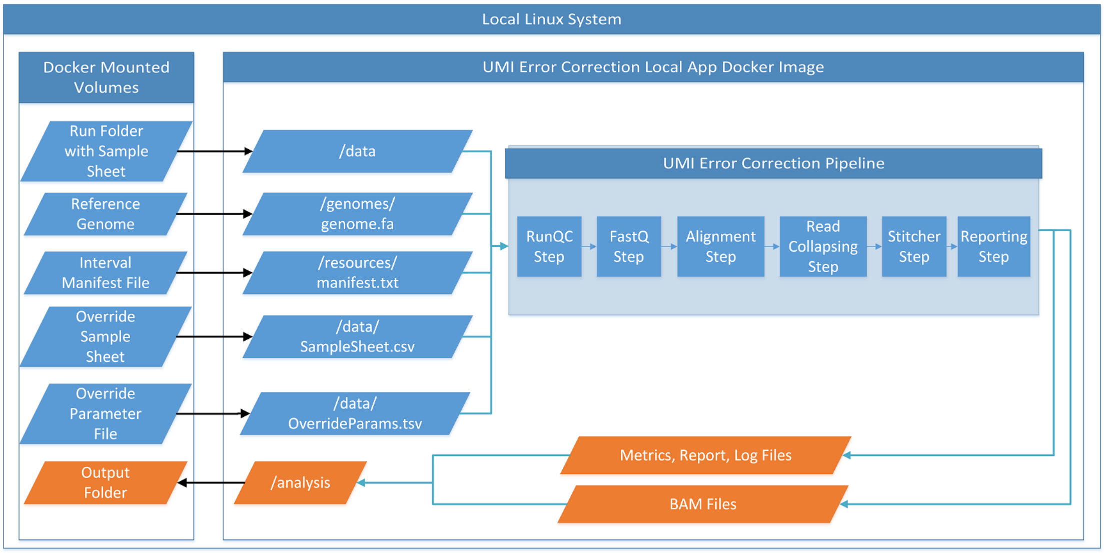
- Perform a full analysis run with the following Docker command, filling in the necessary paths on each volume mount line accordingly and enter the proper User ID in $UID.

  ```
  $ sudo docker run -t --user=$UID -w /analysis \
  -v /etc/localtime:/etc/localtime:ro \
  -v {GENOMES_PATH}:/genomes:ro \
  -v {RUN_PATH}:/data:rw \
  -v {ANALYSIS_OUTPUT}:/analysis:rw \
  -v {MANIFEST_FILE}:/resources/Manifest.txt:ro \
  umi-error-correction-local-app:{AppVersionNumber}
  ```

  The :ro flag equates to read-only access and the :rw flag equates to read-write access. The specified user ($UID) should have access to each system mount location. Verify write access on the :rw flagged mounts and read access on the :ro flagged mounts.
- The container mount points (right side of the colon) need to remain as indicated above or the Pipeline will fail. Customize the left side of the mount based on where files exist on your local system.

Advanced Operation

Figure 2 Advanced UMI Error Correction Local App Docker Image Workflow



- Sample Sheet
  - To specify a sample sheet other than the one contained in the run folder, pass in an additional parameter after the Docker run command with the -s flag and the relative path to the sample sheet within the Docker container. For example:

```
$ sudo docker run -t --user=$UID -w /analysis \
-v /etc/localtime:/etc/localtime:ro \
-v /some/run/analysis:/analysis:rw \
-v /some/run/runfolder:/data:rw \
-v /some/run/genomes:/genomes:ro \
-v /some/path/manifest.txt: /resources/Manifest.txt \
umi-error-correction-local-app:{AppVersionNumber} -s
   /data/SampleSheet_Single_Sample.csv
```

- Override Parameter File
  - The override parameter file and specific parameters are defined in the Advanced Configuration Section.
  - To specify an override parameter file, pass in an additional parameter after the Docker run command with the -p flag and the relative path to the override parameter file within the Docker container. For example:

```
$ sudo docker run -t --user=$UID -w /analysis \
-v /etc/localtime:/etc/localtime:ro \
-v /some/run/analysis:/analysis:rw \
-v /some/run/runfolder:/data:rw \
-v /some/run/genomes:/genomes:ro \
-v /some/path/manifest.txt:/resources/Manifest.txt \
umi-error-correction-local-app:{AppVersionNumber} -p
   /analysis/override.txt
```

- Docker EntryPoint
  - The Docker entry point is the command that the Docker container runs upon execution. By default, the Docker entry point is set to "mono /opt/illumina/NrUmiRunner.exe /data". It is recommended that only advanced users modify the default entry point. An example of the syntax for overriding the default entry point is shown below.

NOTE

Specify mono as the entry point command and append options and parameters to the end of the run command.

```
$ sudo docker run -t --user=$UID -w /analysis \
-v /etc/localtime:/etc/localtime:ro \
-v /some/run/analysis:/analysis:rw \
-v /some/run/runfolder:/data:rw \
-v /some/run/genomes:/genomes:ro \
-v /some/path/manifest.txt:/resources/Manifest.txt \
--entry point="mono" \
umi-error-correction-local-app:{AppVersionNumber} \
/opt/illumina/NrUmiRunner/NrUmiRunner.exe /data -p
   /analysis/override.txt -s /data/SampleSheet_Single_Sample.csv
```

Advanced Configuration: Override Parameters File

The following parameters can be specified in the override parameter file to configure an analysis run. Values are comma-separated when more than one value is needed. Any paths specified in the override parameter file are relative to the Docker container. When override parameters are repeated within the same file, the last name-value pair is used in the analysis.

Table 1 Parameters Specified in the Override Parameters File

Override Parameter	Data Type	Default Value	Description	Example Usage	Available Options
GenomePath	Path	/genomes/genome.fa	Full path to genome file - Used to specify the path within the Docker mount volume to the reference genome.	/genomes/hg19.fa	
ManifestFilePath	Path	/resources/Manifest.txt	Full path to manifest file - Used to specify the path within the Docker mount volume to the manifest file.	/resources/fakemanifest.txt	
AnalysisFolderOverride	Path	/UMI_Error_Correction_Analysis_{DateTime}	Full path to analysis output folder - Used to specify the path within the Docker mount volume to the output analysis folder.	/analysis/OutputFolder1	
SkipSteps	String	Not Used	Skip Steps - Used to specify which steps will not be run in the analysis pipeline.	RunQC, Fastq	RunQC, Fastq, Alignment, ReadCollapsing, Stitcher, Report

Override Parameter	Data Type	Default Value	Description	Example Usage	Available Options
RecoCustomArguments -minMapQuality -minSupportingReads	Strings	-minMapQuality 0 -minSupportingReads 2	Minimum mapping quality - Read Collapsing analysis parameter to filter reads that are considered for collapsing based on mapping quality. Minimum supporting reads - Read Collapsing analysis parameter to filter out read collapsed groups without a minimum number of supporting reads.	RecoCustomArguments -minMapQuality 10 -minSupportingReads 4	-minMapQuality range [0 to 60] -minSupportingReads [0 to 20]
TileFilter	String	Not Used	Tile Filter - Used to run only one tile of the selected data.	1101	Any data tile in the sequencing run
BwaThreadCount	Int	Maximum available **NOTE** 1) The actual number of threads per BWA instance may vary depending on the number of FASTQ pairs, and may be capped at 24 threads when there is only one FASscore. 2) MAPQ scores are known to be influenced by BWA Thread Count in the BWA version 0.7.15 and may influence NoiseAF values. Specify a fixed BwaThreadCount where consistency is desirable.	BWA Thread Count - Used to specify the total number of threads available for BWA during the Alignment step.	16	

Example Override Parameter Files

The columns in an override parameter file are tab-delimited. In the example below, a custom override parameter file is used to perform RunQC, FASTQ generation, and Reporting. The Alignment, ReadCollapsing, and Stitcher steps are skipped. Use the -p option, which allows you to specify a custom parameter override file, to pass a parameter override to the software.

```
AnalysisFolderOverride▸ path▸ /analysis/fastq_20170823-005821_90645
RecoCustomArguments▸ string▸ -minMapQuality 0 -minSupportingReads 2
GenomePath▸ Path▸ /genomes/genome.fa
ManifestFilePath▸ Path▸ /resources/MixDNA_Manifest.txt
SkipSteps▸ string▸ Alignment,ReadCollapsing,Stitcher
```

In the next example, the Alignment, ReadCollapsing, Stitcher, and Report steps are performed after the RunQC and FASTQ generation step, which completed earlier. The analysis folder override location is specified, and custom arguments are provided for the Read Collapsing step.

```
AnalysisFolderOverride▸ path▸ /analysis/docker_20170823-084755_52341
RecoCustomArguments▸ string▸ -minMapQuality 0 -minSupportingReads 2
GenomePath▸ Path▸ /genomes/genome.fa
ManifestFilePath▸ Path▸ /resources/MixDNA_Manifest.txt
SkipSteps▸ string▸ RunQC,FASTQ
```

Due to a dependency on the RunQC and FASTQ generation steps for subsequent steps such as Alignment, the output data from the first example must be staged as input for the second example. Folders and files must exist in the analysis folder named in the override.txt file. This example requires the Fastq folder, which is symbolically linked to the FASTQ output folder from the first example. The second analysis then creates folders for Alignment, Collapsed Reads, Stitched, and Report, which are created in sequential order, and contain the output files from each step.

Staging Data for Running Individual Steps or Skipping Steps

The run folder and corresponding sample sheet are required inputs for running the analysis pipeline from any step. To start from a specific downstream step, such as Alignment, you must first properly stage upstream step data. The analysis folder override parameter needs to reference an already created folder with specific subfolders depending on the desired starting analysis step.

Starting at FASTQ Generation

The only upstream step of FASTQ Generation is RunQC. To skip RunQC and start directly at FASTQ Generation, you do not need to stage any folders, but you must provide the SkipSteps parameter in the override parameter file. Symlinking the Fastq folder to location of the FASTQ files is also recommended.

For example:

```
SkipSteps▸ string▸ RunQC
```

Starting at Alignment

The required upstream step from Alignment is Fastq Generation. To start at Alignment, you must stage the Fastq folder and provide the AnalysisFolderOverride and SkipSteps parameters in the override parameter file.

For example:

```
AnalysisFolderOverride▸ path▸ /analysis/docker_20170823-084755_52341
SkipSteps▸ string▸ RunQC,Fastq
```

The Analysis Folder specified in the AnalysisFolderOverride parameter must contain a folder named Fastq, which must contain FASTQ files corresponding to your sample sheet.

When starting from FASTQ files, the FASTQ files must meet the following requirements.

- Use the template sample sheet (SampleSheet.csv) to create sample sheets when running bcl2fastq to generate FASTQ files. FASTQ file names contain the unique sample ID from the Sample_ID column of the SampleSheet.csv.
- FASTQ file names must contain the Sample ID. By default, bcl2fastq generates FASTQ files with a file name that includes sample names if they are specified in the SampleSheet.csv. When running bcl2fastq independently, leave the Sample_Name field blank so that bcl2fastq generates FASTQ files with the expected file name containing Sample IDs.
- FASTQ files must be named as follows.

  ```
  {Sample_ID}_{SampleNumber}_{LaneNumber}_{ReadNumber}_001.fastq (or
     .fastq.gz)
  ```

  For example:

  ```
  MySample_S1_L001_R1_001.fastq.gz
  MySample_S1_L001_R2_001.fastq.gz
  ```

- Make sure that the memory settings specified in the minimum hardware and software specifications are used when running bcl2fastq.
- It is not required, nor recommended, to merge FASTQ files. Do not use the --no-lane-splitting option in bcl2fastq. The UMI Error Correction Local App analysis pipeline expects separate FASTQ files per sample (Number of FASTQ files = Total Lane Number x two Reads Per Lane).
  - 16 FASTQ files per sample for HiSeq 2500 High Output Run
- Use the following arguments to generate FASTQs using bcl2fastq version 2.19.1.

  ```
  #{BCL2FASTQ_PATH}: Path to bcl2fastq executable
  #{OUTPUT_DIRECTORY_PATH}: Absolute path for output generated fastqs
  #{RUN_FOLDER_PATH}: Absolute path to run folder containing BCLs
  #{SAMPLE_SHEET_PATH}: Absolute path to sample sheet (Typically
  contained within run folder)
  {BCL2FASTQ_PATH) --output-dir {OUTPUT_DIRECTORY_PATH} --runfolder-dir
  {RUN_FOLDER_PATH} --sample-sheet {SAMPLE_SHEET_PATH}
  --barcode-mismatches 1 --adapter-stringency 0.9 --mask-short-adapter-
     reads
  35 --minimum-trimmed-read-length 35
  ```

- FASTQ files must have both Read 1 and Read 2 pairs.

Starting at Read Collapsing

The required upstream step from Read Collapsing is Alignment. To start directly at Read Collapsing, you must stage the Alignment folder as follows and provide the AnalysisFolderOverride and SkipSteps parameters in the override parameter file.

For example:

```
AnalysisFolderOverride▸ path▸ /analysis/docker_20170823-084755_52341
SkipSteps▸ string▸ RunQC,Fastq,Alignment
```

The Analysis Folder specified in the AnalysisFolderOverride parameter must contain a folder named Alignment, which must contain aligned BAM files, and matching index BAI files corresponding to your sample sheet.

Alignment files must be named as follows.

- Aligned BAM files—{Sample_ID}.aligned.bam
- Aligned Index—{Sample_ID}.aligned.bam.bai

For example:

```
MySample.aligned.bam
MySample.aligned.bam.bai
```

Starting at Stitcher

The required upstream step from Stitcher is Read Collapsing. To start directly at Stitcher, you must stage the CollapsedReads folder as follows and provide the AnalysisFolderOverride and SkipSteps parameters in the override parameter file.

For example:

```
AnalysisFolderOverride▸ path▸ /analysis/docker_20170823-084755_52341
SkipSteps▸ string▸ RunQC,Fastq,Alignment,ReadCollapsing
```

The Analysis Folder specified in the AnalysisFolderOverride parameter must contain a CollapsedReads folder, which must contain collapsed BAM files, matching index BAI files, and sample metric files corresponding to your sample sheet.

The CollapsedRead file rules are as follows.

- Collapsed BAM files must be named {Sample_ID}.collapsed.bam
- Collapsed Index files must be named {Sample_ID}.collapsed.bam.bai
- Sample metric files must be named {Sample_ID}_metrics.json

For example:

```
MySample.collapsed.bam
MySample.collapsed.bam.bai
MySample_metrics.json
```

Single Sample Run

The optional -s parameter allows you to specify a custom sample sheet, which can be used to process each sample from a run folder in a single sample parallel processing mode. For example, if a sample sheet has eight samples, it can be duplicated to generate eight minisample sheets, each containing only one sample. Thus, if you have eight nodes on your cluster, you can run all eight samples simultaneously, with a different sample running on each node. Typically, each sample has its own analysis output folder specified in the parameter override file as given in the previous examples. At the end of the run, the individual folders can be organized per sample, and combined programmatically to facilitate further examination of the output and any other downstream analyses.

One caveat to this single sample parallel processing mode is FASTQ generation. Running FASTQ generation on all samples at once is the preferred method for run time performance and optimization. Therefore, Illumina recommends that you conduct an initial analysis using the SkipSteps parameter in the override parameter file, to skip all steps except RunQC and Fastq Generation.

For example:

```
SkipSteps▸ string▸ Alignment,ReadCollapsing,Stitched
```

After this analysis has finished generating all sample FASTQ files, secondary analysis on a per sample basis can be executed as described above. In this analysis, you would skip the RunQC and Fastq step, and start analysis directly at Alignment (See staging FASTQ files for Alignment above).

For example:

```
    SkipSteps▸ string▸ RunQC,Fastq
```

Analysis Methods

Run Quality Control

The Run Quality Control (RunQC) step parses files in the run folder to determine sequencing run quality control information. The following values are calculated and displayed in the report at the end of analysis and in the log file.

- The percentage of reads that pass filter (Percent Reads PF)
  - The percentage of R1 reads that meet Q30 thresholds (Read 1 Q30)
  - The percentage of R2 reads that meet Q30 thresholds (Read 2 Q30)
  - The total number of reads that pass filter (Total Reads PF)
- Instrument Type from Instrument ID
- Comparison of metrics (Read 1 Q30, Read 2 Q30, and Percent Reads PF) to recommended thresholds based on Instrument Type
- Pass/Fail based on comparison

Table 2 Sequencer Specific RunQC Thresholds by Instrument Type

RunQC Metric	HiSeq 2500 Threshold	HiSeq 4000 Threshold	NovaSeq 6000 Threshold
Read 1 Q30	≥80%	≥75%	≥75%
Read 2 Q30	≥75%	≥75%	≥75%
Percent Reads PF	≥80%	≥55%	≥55%

FASTQ Generation

The Fastq step uses the bcl2fastq software to convert BCL files produced on the sequencing instrument into FASTQ files used for read alignment. The version of bcl2fastq (version 2.18 and later) that is used in the analysis pipeline supports UMI usage. For more information, see bcl2fastq Conversion Software on the Illumina Support website.

Alignment

The Alignment analysis step first checks the FASTQ files associated with each sample for correct UMI information. If the FASTQ files do not contain UMI information specific to the TruSight Oncology UMI Reagents, the software does not perform the analysis.

The Burrows-Wheeler Aligner (BWA) uses the Maximal Exact Matches (MEM) algorithm to align DNA reads to hg19 (Homo_sapiens_masked PAR, source UCSC, build hg19). There is a known issue within the BWA version used in the Alignment step, which can produce slightly different results depending on the number of threads used. To ensure consistency between runs, Illumina recommends that you use the same number for the BwaThreadCount parameter in the override parameter file.

This genome is the human reference sequence with the chromosome Y pseudoautosomal regions (PAR) masked (with N's). PAR is present on both the X and Y chromosomes. Because they are identical between chromosomes, reads that map to these regions cannot map uniquely. The masked coordinates are: chrY:10001-2649520 and 59034050-59363566. For reference, the corresponding PAR on chrX are: chrX:60001-2699520 and 154931044-155260560.

The inputs are FASTQ files and the outputs are BAM files and their corresponding BAM index files.

More information about the algorithm can be found on the BWA website (http://bio-bwa.sourceforge.net/).

Read Collapsing

The Read Collapsing analysis step performs an algorithm that uses sets of reads (known as families) with very similar genomic locations and UMI tags to collapse them into representative sequences. This process allows for the accurate removal of duplicate reads without losing the signal of very low frequency sequence variations.

Read Collapsing processes aligned reads from libraries that contain UMIs. These UMIs, along with the positional information in the alignment, are used to group duplicate reads and collapse them into a single consensus aligned read. The resulting reads have higher per-base quality and lower noise from various sources. Read Collapsing also provides various metrics that can be useful for tuning assay development and for quality control when UMIs are involved.

The Read Collapsing step only outputs reads from a successfully collapsed family (single consensus aligned read). The following types of reads are currently discarded from the read collapsed BAM.

- Reads outside of manifest regions
- Split reads
- Large fragment reads

The input files to the Read Collapsing step are an aligned BAM and a BAM index file. The Read Collapsing step creates the following output files.

- Read collapsed BAM
- BAM index file
- Read collapsing metrics JSON file

Note that split reads and large fragment reads (including read pairs mapped to different chromosomes) may be true signals for structural variants (fusions, inversions, and so forth) that are discarded during the read collapsing step. Therefore, for fusion detection, using the raw BWA BAM (plus user preferred mark duplicate program) is recommended instead of the collapsed BAM. For more information on duplex sequencing, refer to the *Indexed Sequencing Overview Guide* (15057455).

The Read Collapsing Step adds the following BAM tags.

- XU: UMI
- XV: Number of Reads in the family
- XW: The number of reads in the duplex-family, or 0 if not a duplex family

Software Error Logs

Software error logs must be configured from the initial command line and captured using the >stdout.txt syntax.

For component-specific logs, refer to the Logs subfolder for the step (applicable for Read Collapsing). Stdout and stderr files are generated in each step-specific output folder.

You can open any of these log files to view error information.

NOTE
The stdout log time uses UTC time zone information.

Analysis Output

Analysis Output Folders

The following output folders and files are generated during an analysis session.

- Fastq Folder
  - <SampleID>_Read#_Lane#.fastq file pairs
  - Fastq_<DateTime>.stderr.txt
  - Fastq_<DateTime>.stdout.txt
- Alignment Folder - BAM with aligned and noncollapsed sequencing reads
  - <SampleID>.aligned.bam
  - <SampleID>.aligned.bam.bai
  - Alignment_<SampleID>_<DateTime>.stderr.txt
  - Alignment_<SampleID>_<DateTime>.stdout.txt
- CollapsedReads Folder - BAM with aligned and collapsed sequencing reads, and intermediate metrics
  - <SampleID>.collapsed.bam
  - <SampleID>.collapsed.bam.bai
  - <SampleID>_metrics.json (intermediate file)
  - ReadCollapserLogs Folder
    - Main Log: ReadCollapserLog.txt
- Stitched Folder – Intermediate BAM and metrics files
  - <SampleID>.error.txt (intermediate file)
  - <SampleID>.nosieAF.json (intermediate file)
  - <SampleID>.stitched.bam (intermediate file)
  - <SampleID>.stitched.bam.bai (intermediate file)
  - Stitcher_<Operation>_<SampleID>_<DateTime>.stdout.txt
  - Stitcher_<Operation>_<SampleID>_<DateTime>.stderr.txt
- Report Folder – Contains the csv formatted report with metrics, software versions, run time and analysis information
  - UMI_Error_Correction_Report<Date><Time>.csv

UMI Error Correction Report

The UMI Error Correction Report (UMI_Error_Correction_Report.csv) is generated for each sample. The title of the report is UMI Error Correction Report, and it contains the following sections and information.

- Analysis Info
  - Analysis Folder
  - Product Version
  - Product Label
  - Report Generation Time
  - Instrument Type (if available)
  - Sequencing Instrument Run QC
  - Read1 Q30
  - Read2 Q30
  - Total Number of Reads Passing Filter
  - Percentage of Reads Passing Filter
  - Recommended Thresholds Based on Instrument Type (if available)
  - Pass/Fail of Thresholds Based on Instrument Type (if available)

- Sample Metrics (see Additional Metrics for further details)

Table 3 Sample Error Correction Report Metrics

Sample Metric Name	Description
Percent Reads With Usable UMIs	The percentage of reads with usable UMIs.
Percent Duplex Families	The percentage of the total number of families that are duplex families.
Percent Families Collapsed	The percentage of the total number of families that are also collapsed.
CV UMI Coverage	The coefficient of variation of the number of families associated with each UMI.
Percent Families Depth >= 5	The number of families with greater than or equal to 5 supporting reads.
Mean Family Depth	Average number of fragments (paired reads) in a family before UMI collapsing.
Mean Family Depth Meets Threshold	Mean Family Depth is greater than or equal to 10 (Derived from ctDNA with TruSight Tumor 170 Workflow).
Median Target Coverage	Median Target Coverage across all base positions.
Median Target Coverage Meets Threshold	Median Target Coverage is greater than or equal to 954 (Derived from ctDNA with TruSight Tumor 170 Workflow).
CV Median Target Coverage	Coefficient of Variation of the Median Target Coverage across all base positions.
Bases with >= 1000x Coverage (%)	Percentage of positions with coverage greater than 1000.
Bases with >= 1500x Coverage (%)	Percentage of positions with coverage greater than 1500.
Bases with >= 2000x Coverage (%)	Percentage of positions with coverage greater than 2000.
Noise AF/Error Rate	Noise Allele Frequency/ Error Rate. Average of all allele frequency values between 0 and 0.05, excluding loci overlapping with known germline mutations from 1000 genome data set.
Noise AF/Error Rate Meets Threshold	Noise Allele Frequency is less than or equal to 7e-5 (Derived from ctDNA with TruSight Tumor 170 Workflow).

- Recommended Quality Thresholds

Table 4 Recommended Quality Thresholds

Metric	Recommended Thresholds* using TruSight Tumor 170 Workflow
Mean Family Depth	>= 10
Median Target Coverage	>= 954
Noise Allele Frequency	<= 7e-5
Read 1 Q30	Based on Instrument Type
Read 2 Q30	Based on Instrument Type
% Read PF	Based on Instrument Type

*Thresholds are derived from cfDNA inputs. Sheared gDNA inputs may not meet these thresholds.

  - For more information on recommended thresholds, see *Frequently Asked Questions* on page 26.
- Software Versions
  - UMI Pipeline Version
  - Mono Version
  - BWA Version
  - SAMtools Version
  - ReCo Version
  - Stitcher Version
  - Dotnet Core Version
- Analysis Runtime
  - RunQC Step
  - Fastq Step
  - Alignment Step
  - Read Collapsing Step
  - Stitcher Step
  - Total Runtime
- Resource Information
  - Genome and Manifest Files
  - File Type, Checksum (md5) and File Path

Additional Metrics Information

The file name for the Read Collapsing Metrics JSON file is <SampleID>_metrics.json. This JSON file contains the following read collapsing metrics.

Table 5 Read Collapsing Metrics

Metric Name	Description	Calculation
MeanFamilyDepth	Average number of fragments (paired reads) in a family before UMI collapsing.	SUM(Reads in each family)/ the number of families after correction, collapsing, and filtering on supporting reads.
PercentFamiliesCollapsed	The percentage of the total number of families that are also collapsed.	100 * FamiliesWritten / (FamiliesCreated - FamiliesContextuallyCorrected - FamiliesShifted - DuplexFamilies)
PercentDuplexFamilies	The percentage of the total number of families that are duplex families.	100 * DuplexFamilies / (FamiliesCreated - FamiliesContextuallyCorrected - FamiliesShifted - DuplexFamilies)
PercentReadsWithUsableUmis	The percentage of reads with usable UMIs.	100 * (ReadsOnTarget- FilteredOutReads - ReadsDiscardedInvalidUmis) / (ReadsOnTarget - FilteredOutReads)
CvUmiCoverage	The coefficient of variation of the number of families associated with each UMI.	Math.Sqrt (UmiFamilyCounts.Average(v => Math.Pow(v - avg, 2))) / UmiFamilyCounts.Average()
ReadsOnTarget	The number of raw reads that overlap with manifest intervals.	Incremented by one for each read
FilteredOutReads	The number of raw reads that are discarded before creating families, including families that are unpaired, mate unmapped, without cigar string, or low map quality.	Incremented by one for each read
SVCandidateReads	The number of raw reads that are considered as structural variation candidates, including fragment that exceeds max size specified or mapped onto the different chromosomes.	Incremented by one for each read
FamiliesCreated	The total number of families created.	Incremented by one for each family
FamiliesContextuallyCorrected	The number of families that are merged into other families based on contextual UMI correction.	Incremented by one for each family
FamiliesShifted	The number of families merged into other families based on proximity with matching UMIs.	Incremented by one for each family
FamiliesWritten	The number of families written out in the form of a condensed read pair.	Incremented by one for each family
DuplexFamilies	The number of families that meet the criteria for cross-collapsing.	Incremented by one for each family
FamiliesDiscarded	The number of families that are discarded during correction.	Incremented by one for each family

Metric Name	Description	Calculation
FamiliesDiscardedInvalidUmis	The number of invalid families that are discarded, including uncorrectable UMIs.	Incremented by one for each family
ReadsDiscardedInvalidUmis	The number of invalid reads that are discarded, including uncorrectable UMIs.	
ReadsInDiscardedFamilies	The number of reads that are thrown away when discarding families.	Incremented by total reads in each discarded family
ReadsWithValidOrCorrectedUmi	The number of reads with a particular UMI after UMI correction, prior to collapsing and filtering on supporting reads.	Incremented by two for each read
FamiliesWithValidOrCorrectedUmi	The number of families with a particular UMI after correction, collapsing but before filtering on supporting reads.	Incremented by one for each family
NumSupportingFragments	An array of counts for how many families have this many reads supporting it. Calculated after UMI correction, collapsing, and filtering on supporting reads.	Incremented by one for each read per family
FragmentSizeHistogram	The histogram of fragment size of condensed (consensus) families.	Incremented by one for each read

The file name for the Error Rate Metrics JSON file is <SampleID>_noiseAF.json. This JSON file contains the following metrics.

Table 6 Error Rate Metrics

Name	Description	Mathematical definition
Total_DP	Total Depth of Fragment Coverage	Sum of depth of fragment coverage (nA+nC+nG+nT+nIndel)
Total_AD	Total Depth of Alternative Allele	Sum of alternative allele Number of alternative allele = Min (total counts different from the reference base, total counts same as the reference base). Note: when a patient whose genotype is the homozygous variant, total alternative depth is total counts same as the Reference base.
NoiseAF	Noise Allele Frequency/Error Rate	Average of all allele frequency values between 0 and 0.05, excluding loci overlapping with known gdata set mutations from 1000 genome data set.
MTC	Median Target Coverage across all base positions	Median of depth of fragment coverage (nA+nC+nG+nT+nIndel) across all base positions.
CV_MTC	Percentage Coefficient of Variation of Target Coverage across all base positions	100 * median absolute deviation of depth of fragment coverage (nA+nC+nG+nT+nIndel) / Median of depth of fragment coverage (nA+nC+nG+nT+nIndel).
PCT_1000X	Parameter insensitive to input manifest	(Number of positions with depth of fragment coverage >1000)/(Total number of positions) *100%
PCT_1500X	Parameter insensitive to input manifest	(Number of positions with depth of fragment coverage >1500)/(Total number of positions) *100%
PCT_2000X	Parameter insensitive to input manifest	(Number of positions with depth of fragment coverage >2000)/(Total number of positions) *100%

Uninstall UMI Error Correction Local App

The following steps remove all assets, Docker images, and Docker containers (if you performed any analyses).

1 Make sure that there are no active runs being performed (there should not be any listed containers containing the image: umi-error-correction-local-app).

```
$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

2 If there is an active Docker container associated with the UMI_Error_Correction_Local_App image, and it is no longer needed, you can stop its execution with the following command.

```
# using the container id, found in the above command:
$ sudo dockers stop <container_id>
```

3 Remove all previous Docker containers associated with the umi-error-correction-local-app image.

```
# similar to the above, identify the container id:
$ sudo docker ps -a|grep umi-error-correction-local-app

# then remove each container:
$ sudo docker rm $(sudo docker ps -a|grep umi-error-correction-local-
   app|awk '{print $1}')
```

4 Remove the Docker image.

```
$ sudo docker rmi -f umi-error-correction-local-app:{AppVersionNumber}
```

Docker should now be completely cleaned from the image and previous runs.

5 Remove downloaded assets.

```
$ rm {DOWNLOAD_PATH}/UMI_Error_Correction_Local_App_{AppVersionNumber}.tgz
   {DOWNLOAD_PATH}/UMI_Error_Correction_Local_App_Resources_
   {AppVersionNumber}.gz
```

Frequently Asked Questions

- How many samples can a specific instrument multiplex using UMIs?
  - Recommended rule is for every 100 kb of panel, 50–60 million PF clusters are needed.
  - HiSeq 2500
    - 2B clusters / (60M * (PanelSizeinMB/100Kb)) = Multiplex Samples
  - HiSeq 4000
    - 2.5B clusters / (60M * (PanelSizeinMB/100Kb)) = Multiplex Samples

  Example: (1 Mb panel): 2.5B clusters / (60M * (1MB/100Kb)) = 4 Samples
- What might happen if the amount of DNA is too high or low or has poor fragmentation?
  - If the DNA input is too high, the MTC will be very high, leading to small family size and poor error correction. This may cause an ‘outside of threshold’ result for the mean family size >= 10 metric.
  - If the DNA input is too low, the MTC will be very low, leading to over sequencing (ie large family size) and may cause an ‘outside of threshold’ result for the MTC >= 954 metric. Over sequencing may lead to increased inherent error rate.

  - For cfDNA applications, no fragmentation is needed (they are naturally fragmented). For genomic DNA samples, it is recommended that you fragment to median size ~160 bp to mimic cfDNA size range.

- Interpretation of results: What conclusions can be drawn from the following scenarios? Is the data usable? What can I do to correct any observed issues?

Meets Median Target Coverage Threshold	Meets Mean Family Depth Threshold	Meets Noise Allele Frequency Threshold	Conclusion
Yes	Yes	Yes	Data are usable for ultralow AF detection (~0.4%).
Yes	Yes	No	Unlikely scenario. If observed, the sample may have significant DNA damage before library prep.
Yes	No	Yes	Slightly insufficient sequencing however, data might still be good to use for variant detection.
Yes	No	No	Insufficient sequencing for DNA input amount. Family size is too small to effectively reduce error rate.
No	No	Yes	Low DNA input or problem in library conversion efficiency. Slightly under sequencing. Not recommended for ultralow AF detection due to low MTC.
No	Yes	Yes	Low DNA input or problem in library conversion efficiency. Sufficient sequencing. Not recommended for ultralow AF detection due to low MTC.
No	Yes	No	Low DNA input or problem in library conversion efficiency. Sufficient sequencing but sample may have significant DNA damage before library prep.
No	No	No	Low DNA input or problem in library conversion efficiency. Under sequencing. Family size is too small to effectively reduce error rate.

NOTE

  - Median Target Coverage Threshold is dependent on panel.
  - Mean Family Depth and Noise Allele Frequency Thresholds are independent of panel but are source material dependent. Sheared gDNA inputs may not meet thresholds.

- How much Disk Space is recommended for End-to-End Analysis?
  - HiSeq 2500/HiSeq 4000 – 1 TB

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

Revision History

Document	Date	Description of Change
Document # 1000000037270 v00	April 2018	Initial Release.

Technical Assistance

For technical assistance, contact Illumina Technical Support.

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

Illumina Customer Support Telephone Numbers

Region	Toll Free	Regional
North America	+1.800.809.4566	
Australia	+1.800.775.688	
Austria	+43 800006249	+43 19286540
Belgium	+32 80077160	+32 34002973
China	400.066.5835	
Denmark	+45 80820183	+45 89871156
Finland	+358 800918363	+358 974790110
France	+33 805102193	+33 170770446
Germany	+49 8001014940	+49 8938035677
Hong Kong	800960230	
Ireland	+353 1800936608	+353 016950506
Italy	+39 800985513	+39 236003759
Japan	0800.111.5011	
Netherlands	+31 8000222493	+31 207132960
New Zealand	0800.451.650	
Norway	+47 800 16836	+47 21939693
Singapore	+1.800.579.2745	
Spain	+34 911899417	+34 800300143
Sweden	+46 850619671	+46 200883979
Switzerland	+41 565800000	+41 800200442
Taiwan	00806651752	
United Kingdom	+44 8000126019	+44 2073057197
Other countries	+44.1799.534000	

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

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

Document # 1000000037270 v00

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

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