Twist FlexPrep UHT Library Preparation Kit: Sample Demultiplexing Guide

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

TABLE OF CONTENTS

INTRODUCTION

STANDARD ILLUMINA INDEXING WORKFLOW

In next-generation sequencing (NGS) methodologies, sample multiplexing is indispensable for maximizing throughput and cost-effectiveness. Illumina library preparation incorporates barcodes on one or both sides of the library molecule to facilitate concurrent sequencing of multiple samples within a single run. The process, termed demultiplexing, computationally assigns sequencing reads to their sample of origin based on their respective barcode information.

The standard Illumina-style placement of barcodes is outside of the library insert sequence. Illumina uses dedicated indexing reads, termed i5 and i7, to determine the sequence of the barcodes (**Figure 1**).

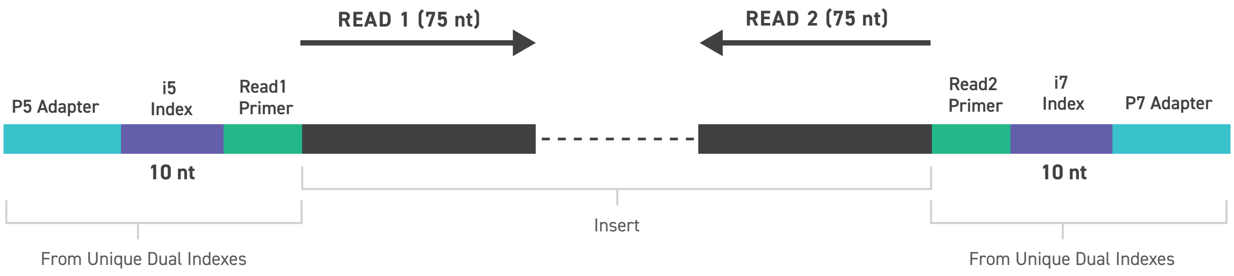

Figure 1. Traditional NGS library structure for dual-index sequencing.

Following sequencing on Illumina platforms, resultant binary base calling (.BCL) files encapsulate the raw sequencing data. The typical workflow involves using the Illumina software suite to convert BCL files into FASTQ files, while concurrently demultiplexing samples using the i5 and i7 indexes. This generates one FASTQ file per sample.

TWIST FLEXPREP INDEXING WORKFLOW

The Twist FlexPrep UHT Library Preparation Kit leverages a set of 12 inline barcodes to allow for pooling and purification of samples post-ligation, thereby enhancing downstream throughput (**Figure 2**).

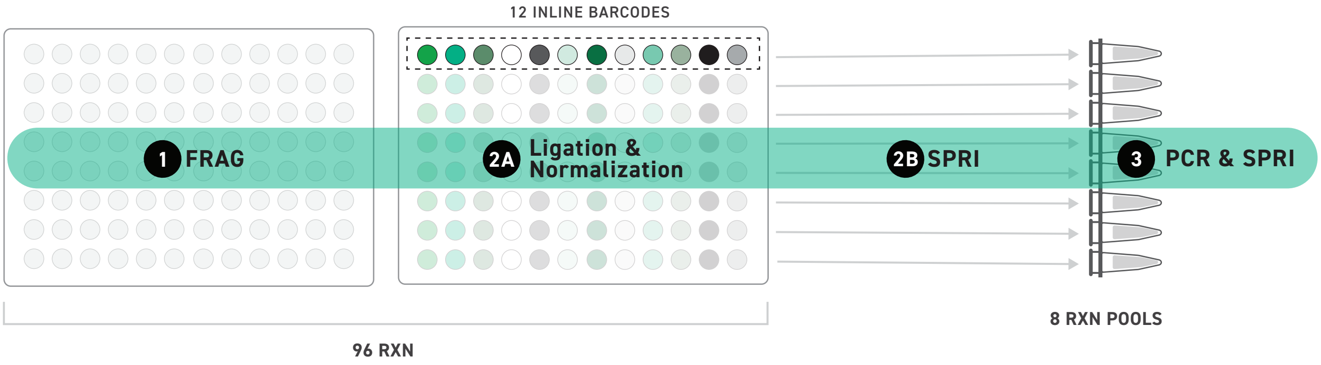

Figure 2. Twist FlexPrep Library Preparation Workflow.

Each additional Twist FlexPrep barcode is inline as part of read 1 (R1) and read 2 (R2). Existing Illumina BCL conversion and sample demultiplexing software (e.g., bcl2fastq2 and BCLconvert) are not currently equipped to handle the dual-step demultiplexing required for samples with barcodes at the start of template reads. Consequently, Illumina software treats Twist FlexPrep libraries as conventional dual-index libraries. The software requires an additional step to segregate the library sequences into their respective 12 individual FASTQ files while maintaining the shared i5 and i7 (**Figure 3**).

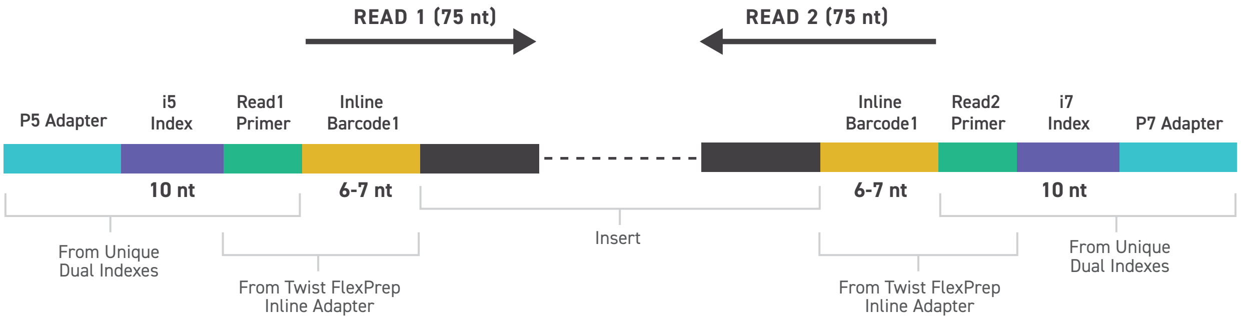

Figure 3. Twist FlexPrep Final Library Structure for Post-Ligation Pooling.

To generate a FASTQ for each sample when processing libraries prepared with the Twist FlexPrep UHT Library Preparation Kit, a two-step approach is needed for demultiplexing and BCL conversion (**Figure 4**). First, the BCLs are converted to FASTQ while demultiplexing using the UDI barcodes (up to 400 options) with standard Illumina software. Second, the inline sample barcode (12 options) is demultiplexed using the fgbio DemuxFastqs opensource software to produce one FASTQ file per sample.

This document outlines an example workflow for processing 1 Illumina demultiplexing (UDI) FASTQ into 12 inline sample FASTQs based on barcodes at the start of the first (R1) and second (R2) sequencing reads.

NOTE: FlexPrep libraries can be pooled with other dual-indexed libraries on the same flow cell for sequencing.

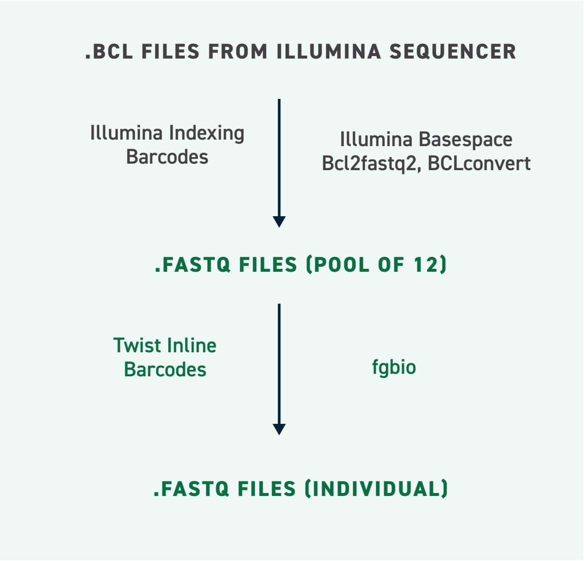
.BCL FILES FROM ILLUMINA SEQUENCER
Illumina Indexing Barcodes
Illumina Basespace Bcl2fastq2, BCLconvert
.FASTQ FILES (POOL OF 12)
Twist Inline Barcodes
fgbio
.FASTQ FILES (INDIVIDUAL)

Figure 4. High-Level Sequencing Demultiplexing Workflow.

REFERENCED SOFTWARE PACKAGES

The following software packages are used in the examples within this document:

PACKAGE	VERSION	LICENSE	URL
fgbio	2.0.2	MIT	https://github.com/fulcrumgenomics/fgbio/releases/tag/2.0.2

NOTE: Input of parameters for each tool within each package will affect analysis results. We recommend that you start by running individual tools with the "-h" option to view a full list of parameters that you can tune.

TWIST FLEXPREP INLINE BARCODE DEMUX

STEP 1. GETTING STARTED

Direct Download

- Go to the latest release page of fgbio
- Download the fgbio.jar file
- Open up the command terminal and run the following command to see the tools available:

```
java -jar fgbio.jar
```

- To run fgbio, you will need Java 8 (aka Java 1.8) or later installed.
- To see which version of Java is currently installed, run the following in a command terminal:

```
java -version
```

If the reported version on the first line starts with 1.8 or higher, you are all set.
For isolation and versioning purposes, Twist uses Docker with the following container:

Using Docker

- Install Docker if it is absent in your system
- Pull the fgbio image from the Biocontainers repository and run it through Docker:

```
docker pull quay.io/biocontainers/fgbio:2.0.2--hdfd78af_0
docker run  quay.io/biocontainers/fgbio:2.0.2--hdfd78af_0
```

STEP 2. GENERATING A SAMPLE SHEET

In lieu of a sample sheet, a simple CSV file is provided with the necessary barcodes below, make a copy of it and use it for Step 2, <Sample>.csv:

```
Sample_Id,Sample_Name,Sample_Barcode
Sample_S1,Well_1, GCCATAGCCATA
Sample_S2,Well_2, TCTGGTTCTGGT
Sample_S3,Well_3, TGGCTTTGGCTT
Sample_S4,Well_4, AACACTAACACT
Sample_S5,Well_5, TCAGGATCAGGA
Sample_S6,Well_6, TGGTCCTGGTCC
Sample_S7,Well_7, CAACTGCAACTG
Sample_S8,Well_8, CGGACCCGGACC
Sample_S9,Well_9, ATCGAGATCGAG
Sample_S10,Well_10, ATGGTGATGGTG
Sample_S11,Well_11, CTTAAGCTTAAG
Sample_S12,Well_12, GAGTGCGAGTGC
```

STEP 3. DEMULTIPLEX POOL OF 12 FASTQS

The FASTQ files for each given pool of 12 samples after Illumina demultiplexing are used as input to produce a per-sample FASTQ file. The demultiplexing must be performed for each pool of 12 samples independently.

NOTE: If downsampling is required, we recommend doing it at the per-sample FASTQ level after this demultiplexing step. Performing downsampling at this juncture may risk allocating insufficient reads to individual samples.

The fgbio DemuxFastqs tool performs the sample demultiplexing by comparing the identified sample barcode from read structures to expected barcodes provided in metadata to assign each read to a sample. Reads that do not match any sample within the specified error tolerance will be placed in the “unmatched” file.

Example invocation:

```
java -jar fgbio.jar DemuxFastqs \
  --metadata <Sample>.csv \
  --min-mismatch-delta 2 \
  --max-mismatches 1 \
  --read-structures 6B2S+T 6B2S+T \
  --inputs <Fastq_r1>.fastq.gz <Fastq_r2>.fastq.gz \
  --output output \
  --output-type Fastq \
  --metrics <metrics>.tsv
```

Notes:

- The read structure parameter defines the Twist FlexPrep specific read structure, as follows (see Figure 3):
  - 6B2S+T: Read 1 structure - 6 bases for the Barcode; 2 bases to Skip (phasing and T/A ligation junction); rest of the read is Template
  - 6B2S+T: Read 2 structure - 6 bases for the Barcode; 2 bases to Skip (phasing and T/A ligation junction); rest of the read is Template
  - See the documentation for read structures via the fgbio Read Structures webpage
- The output of the above is in FASTQ format and will contain only template bases. fgbio will trim the inline 6/7-nt barcode as well as the single base that correspond to the A/T ligation junction.
- To output BAM format use `--output-type Bam`. The BAM file will contain the sample barcode per read in the BC tag. Both BAM and FASTQs may be output with `--output-type FastqAndBam`.

The full set of supported command line options can be found in the documentation for fgbio’s DemuxFastqs.

The sample sheet provided to fgbio’s DemuxFastqs can be validated online via the fgbio DemuxFastqs sample sheet validation webpage. Validating the sample sheet can help avoid cases where an incorrect sample sheet causes a job to fail.

An example output directory structure is found in the Appendix.

STEP 4. UNDERSTANDING OUTPUT METRICS

The output metrics file will contain per-sample demultiplexing statistics; see the documentation for fgbio's SampleBarcodeMetric.

STEP 5. TROUBLESHOOTING (OPTIONAL)

The fgbio DemuxFastqs tool fails with an out of memory exception

The tool will fail with the following exception:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
```

In this case, the amount of memory allocated to the tool needs to be increased. This is done with the -Xmx Java Virtual Machine (JVM) option for the heap space. For example, to specify 16 gigabytes of memory:

```
fgbio -Xmx16g DemuxFastqs
```

There are few or no reads in the FASTQs for the sample barcodes and the unmatched FASTQs contain most of the reads

This may occur if the sample barcodes were specified incorrectly to the fgbio DemuxFastqs tool. Double-check to ensure that the sample barcodes are correct.

Sample barcode not found when demultiplexing the samples

The fgbio DemuxFastqs tool will search for the column with the name Sample_Barcode to specify the inline read sample barcode for each sample. If that column is not found, the tool will fail with the following error:

```
requirement failed: Sample barcode not found in column 'Sample_Barcode'
for sample id '<sample-name>'.
```

Please make sure that the column with the name Sample_Barcode is found in the sample metadata. The column used for the sample barcode may be changed with --column-for-sample-barcode <value> command line option.

APPENDIX

EXAMPLE OUTPUT DIRECTORY STRUCTURE FROM FGBIO DEMUXFASTQS

An example file and directory structure follows

```
/path/to/outputs/samples/12_plex-1
├── <metrics>.tsv
├── Sample_S1_Well_1_R1.fastq.gz
├── Sample_S1_Well_1_R2.fastq.gz
├── Sample_S2_Well_2_R1.fastq.gz
├── Sample_S2_Well_2_R2.fastq.gz
├── Sample_S3_Well_3_R1.fastq.gz
├── Sample_S3_Well_3_R2.fastq.gz
├── Sample_S4_Well_4_R1.fastq.gz
├── Sample_S4_Well_4_R2.fastq.gz
├── Sample_S5_Well_5_R1.fastq.gz
├── Sample_S5_Well_5_R2.fastq.gz
├── Sample_S6_Well_6_R1.fastq.gz
├── Sample_S6_Well_6_R2.fastq.gz
├── Sample_S7_Well_7_R1.fastq.gz
├── Sample_S7_Well_7_R2.fastq.gz
├── Sample_S8_Well_8_R1.fastq.gz
├── Sample_S8_Well_8_R2.fastq.gz
├── Sample_S9_Well_9_R1.fastq.gz
├── Sample_S9_Well_9_R2.fastq.gz
├── Sample_S10_Well_10_R1.fastq.gz
├── Sample_S10_Well_10_R2.fastq.gz
├── Sample_S11_Well_11_R1.fastq.gz
├── Sample_S11_Well_11_R2.fastq.gz
├── Sample_S12_Well_12_R1.fastq.gz
├── Sample_S12_Well_12_R2.fastq.gz
├── unmatched_R1.fastq.gz
└── unmatched_R2.fastq.gz
```

HAVE FURTHER QUESTIONS?

For additional support please contact Twist Bioscience's support team at **customersupport@twistbioscience.com.**

These products are for research use only, and subject to additional use restrictions as set forth in Twist's Supply Terms and Conditions: www.twistbioscience.com/supply-terms-and-conditions