NEXT-GEN SEQUENCING

# SMARTer™ Human scTCR Demultiplexer Guide

## I. Introduction

The SMARTer Human scTCR Demultiplexer is a piece of software for use only with the SMARTer® Human scTCR a/b Profiling Kit. This software uses in-line indexes incorporated into the paired reads to demultiplex your samples and generate an output file containing sorted read data. These in-line indexes have been hard-coded in the software.

The SMARTer Human scTCR a/b Profiling Kit generates a library in which unique in-line indexes added to the cDNA serve as cell barcodes and enable sample pooling prior to library preparation and sequencing. The library produces paired reads where the in-line index is encoded in six nucleotides of Read 2 (bases 33–38), followed by the sequence of interest. For more information on the SMARTer Human scTCR a/b Profiling Kit, please visit www.takarabio.com/hscTCR_kit.

**NOTE:** If you know you have Java 1.7 or later installed and are only looking for the command line arguments, please see Section IV.B.

## II. Before You Begin

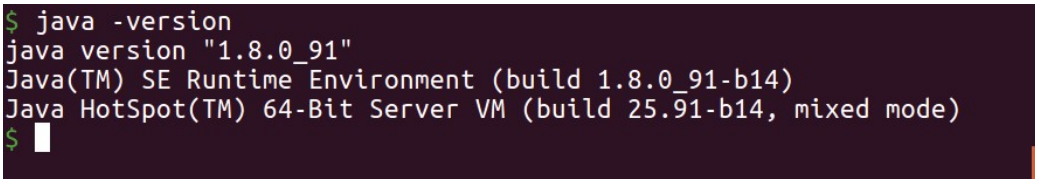
### A. Input files

Input files are paired reads where Read 2 contains an in-line barcode sequence. The SMARTer Human scTCR Demultiplexer works with either gzipped files (**.gz** extension) or uncompressed files (commonly with **.fq** or **.fastq** extensions).

### B. Verifying Java and confirming operating system

The SMARTer Human scTCR Demultiplexer is written in Java and is compatible with any operating system where Java 1.7 or newer is installed. We have tested and confirmed operation of the SMARTer Human scTCR Demultiplexer on Linux (Ubuntu 16.10), Windows (7 and 10), and Mac (OS X El Capitan) systems.

After opening a terminal window, verify that you have the correct version of Java installed by typing the `java -version` command into your terminal window (see Section IV.A for instructions on opening a terminal window):



```
$ java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
$ 
```

**Figure 1. How to verify the Java version of your OS.** After typing “java -version” into your terminal, you should see a version number displayed. Verify that the first two numbers are 1.7 or higher.

### C. Sample data

We have provided sample data in the zip archive (within the folder named "Plate 3") that you can use to confirm operation of the demultiplexing software. The paired-end sample data contains 12 libraries, each consisting of a pooled library with all eight in-line indexes. This representative dataset was acquired with a library synthesized using the SMARTer Human scTCR a/b Profiling Kit and sequenced on an Illumina® MiSeq® instrument with paired-end, 2 x 300 base pair reads. The data are contained in twenty-four files named:

`N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz` (from Read 1) and

`N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz` (from Read 2)

through

`N24-plate3-12-RNA_S24_L001_R1_001.fastq.gz` (from Read 1) and

`N24-plate3-12-RNA_S24_L001_R2_001.fastq.gz` (from Read 2)

## III. Installing SMARTer Human scTCR Demultiplexer

The software is available for download as a zip file from takarabio.com/hscTCR_demultiplexer. Unzip the archive in a convenient directory. The software is packaged as an executable Java jar file. No additional steps are necessary to install the software.

## IV. Running the SMARTer Human scTCR Demultiplexer

The SMARTer Human scTCR Demultiplexer is written as a command line interface tool and is executed from within a terminal window.

### A. Opening a terminal window

- **Mac:** The Terminal application is typically found under **Applications > Utilities > Terminal**
- **Linux (Ubuntu):** Use the keyboard shortcut **[Ctrl][Alt][T]**.
  Alternatively, you can find the Terminal by opening the Dash (upper left on most desktops), typing "`terminal`" and selecting the Terminal application.
- Windows 10: Click the **Start** button and type "`cmd`" in the run box.
- Alternatively, select Command Prompt from **Start > Windows Systems > Command Prompt**

> **NOTE:** On older Windows systems, the path to Command Prompt is **Start > Program Files > Accessories > Command Prompt**

### B. Running the demultiplexing command with a single pair of fastq files

The command to demultiplex your sequencing run is:

```
java -jar /path/to/TCR_Demux.jar /path/to/Read1 /path/to/Read2
```

> **NOTE:** On Mac and Linux systems ~/ is not expanded automatically by Java.
>
> Use the full path, i.e.
>
> ```
> /home/takara/TCR_Demux.jar /N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz
> ```
>
> Do not use
>
> ```
> ~/TCR_Demux.jar /N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz
> ```

The table on the next page provides a brief explanation of the command.

| Command component | Explanation |
|---|---|
| `java -jar` | System command to execute a Java jar file |
| `/path/to/TCR_Demux.jar` | Full system path to the SMARTer Human scTCR Demultiplexer jar file |
| `path/to/Read1` | Full system path to Read 2 fastq file |
| `path/to/Read2` | Full system path to Read 1 fastq file |

**NOTE:** As an alternative to typing the demultiplex command, you can drag and drop files into the command window. If you'd like to avoid typing out the full file paths, you can drag a file onto the terminal window to automatically add each file path to the end of the current command line. To use this method to fill out the full command, use the following steps:

1. Open the terminal window and the folder containing the `TCR_Demux.jar` file and your read data
2. Type `java -jar`
3. Press space bar
4. Drag and drop the `TCR_Demux.jar` file into the terminal window (you should now see a full path to the executable file)
5. Press space bar
6. Drag and drop the Read 1 file into the terminal window (you should see the full path for the **fastq** file)
7. Press space bar
8. Drag and drop the Read 2 file into the terminal window (you should see the full path for the **fastq** file)

## Example command

Here, we provide an example command run on Linux with the zip archive extracted into our home folder (**home/takara/**). The following command will demultiplex the example data included in the archive:

```
java -jar
/home/takara/TCR_Demux/TCR_Demux.jar
/home/takara/TCR_Demux/N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz
/home/takara/TCR_Demux/N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz
```

**NOTE:** The command is a single line.

## Running the demultiplexing command with a directory of pairs fastq files

The command to demultiplex your sequencing run is:

```
java -jar /path/to/TCR_Demux.jar /path/to/directory
```

**NOTE:** All files in the directory that have the following extensions will be used: **.gzip; .gz; .fastq;** and **.fq**. Program expects the files to be paired and differ in name, (e.g., by 'R1' or 'R2' designation).

# V. Examining the Demultiplexing Command Output

## A. Output files for a single pair of fastq files

The demultiplexing command always produces 18 files for every pair of fastq files, even if no sequences are placed in a file (e.g., if you are analyzing data from four pooled samples, there will still be 18 files produced by the software). The original data files are maintained in their original directory, but a second directory will be created that includes the newly generated files. This directory will be named "OutputFiles". Four kinds of files are produced by the demultiplexing command:

- **Read 1:** There are eight "Read 1" output files that are named with the barcode name added to the beginning of the original file name and the extension **.fastq** added to the end. These files now contain sequencing reads that are usable for downstream analyses.

  e.g., `Barcode_A_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq`

- **Read 2:** There are eight "Read 2" output files that are named with the barcode name added to the beginning of the original file name and the extension **.fastq** added to the end. These files now contain sequencing reads that are usable for downstream analyses.

  e.g., `Barcode_A_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq`

- **Undetermined files:** There are two "UNDETERMINED" files, one associated with Read 1 and one associated with Read 2. When the software cannot match the nucleotides in Read 1 or Read 2 to one of the eight expected barcodes, it places those reads in the:

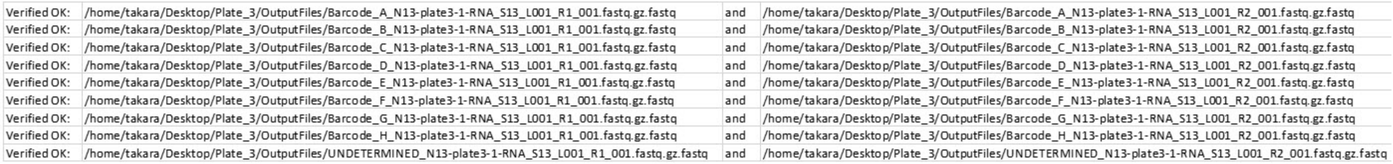
  `UNDETERMINED_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq` and

  `UNDETERMINED_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq` file-pair.

- **Summary file:** A summary file is generated for each run. This tab-delimited .txt file is named for the date and time the software was run (e.g., `Mon Jun 05 13_32_45 PDT 2017_Results Summary.txt`). Immediately upon initiating the command, the summary file is generated with the date, the location of the files to be analyzed, and the location of the "OutputFiles" directory (example below).

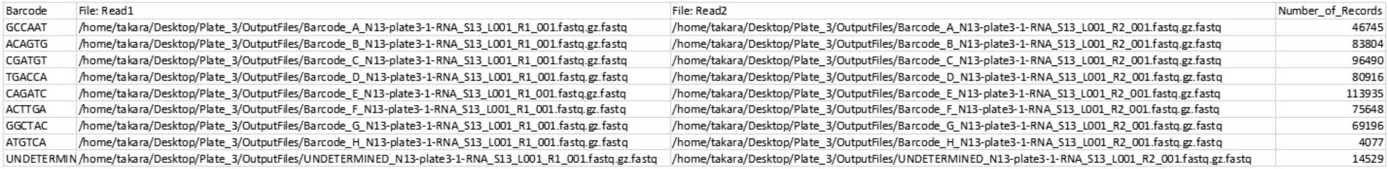
| |
|---|
| Summary: Mon Jun 05 13_32_45 PDT 2017 |
| |
| This summary file contains results based on the following request: |
| The folder pointing to the reads was: /home/takara/Desktop/Plate_3 |
| A single base mismatch in the barcodes was permitted. |
| |
| DIR: /home/takara/Desktop/Plate_3/OutputFiles was created successfully. |
| |
| Running... |

Subsequently, there will be a log showing that the output **.fastq** files were generated properly and each file is listed (example below).



| | | | |
|---|---|---|---|
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_A_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_A_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_B_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_B_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_C_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_C_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_D_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_D_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_E_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_E_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_F_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_F_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_G_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_G_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_H_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_H_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |
| Verified OK: | /home/takara/Desktop/Plate_3/OutputFiles/UNDETERMINED_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | and | /home/takara/Desktop/Plate_3/OutputFiles/UNDETERMINED_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq |

Finally, there will be a log of how many reads were placed into each of the newly generated files (example below).



| Barcode | File: Read1 | File: Read2 | Number_of_Records |
|---|---|---|---|
| GCCAAT | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_A_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_A_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 46745 |
| ACAGTG | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_B_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_B_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 83804 |
| CGATGT | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_C_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_C_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 96490 |
| TGACCA | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_D_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_D_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 80916 |
| CAGATC | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_E_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_E_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 113935 |
| ACTTGA | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_F_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_F_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 75648 |
| GGCTAC | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_G_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_G_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 69196 |
| ATGTCA | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_H_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/Barcode_H_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 4077 |
| UNDETERMIN | /home/takara/Desktop/Plate_3/OutputFiles/UNDETERMINED_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq | /home/takara/Desktop/Plate_3/OutputFiles/UNDETERMINED_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq | 14529 |

## B. Output files for a directory of pairs of fastq files

The demultiplexing command always produces 18 files for each pair of fastq files. In the example data set, the different fastq pairs are designated by "N13—N24" and "S13—S24" in the fastq file name. The original data files are maintained in their original directory, but a second directory will be created that includes the newly generated files. This directory will be named "OutputFiles". Four kinds of files are produced by the demultiplexing command:

- **Read 1:** There are eight "Read 1" output files for each fastq pair that are named with the barcode name added to the beginning of the original file name and the extension **.fastq** added to the end. These files now contain sequencing reads that are usable for downstream analyses. If all 12 possible fastq files were used, then the output file will contain 96 "Read 1" output files ready for further analysis.

  e.g., `Barcode_A_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq`

- **Read 2:** There are eight "Read 2" output files for each fastq pair that are named with the barcode name added to the beginning of the original file name and the extension .fastq added to the end. These files now contain sequencing reads that are usable for downstream analyses. If all 12 possible fastq files were used, then the output file will contain 96 "Read 2" output files ready for further analysis.

  e.g., `Barcode_A_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq`

- **Undetermined files:** There are two "UNDETERMINED" files, one associated with Read 1 and one associated with Read 2. When the software cannot match the nucleotides in Read 1 or Read 2 to one of the eight expected barcodes, it places those reads in the:

  `UNDETERMINED_N13-plate3-1-RNA_S13_L001_R1_001.fastq.gz.fastq` and

  `UNDETERMINED_N13-plate3-1-RNA_S13_L001_R2_001.fastq.gz.fastq` file-pair.

  If all 12 possible fastq files were used, there will be 12 sets of "UNDETERMINED" files.

- **Summary file:** A summary file is generated for each run. This tab-delimited .txt file is named for the date and time of the running of the software (e.g., `Mon Jun 05 13_32_45 PDT 2017_Results Summary.txt`). Immediately upon initiating the command, the summary file is generated with the date, the location of the files to be analyzed, and the location of the "OutputFiles" directory (example above). Subsequently, there will be a log that the output **.fastq** files were generated properly, and each file is listed (example above). Finally, there will be a log of how many reads were placed into each of the newly generated files (example above). The last two operations will be repeated for all sets of **.fastq** files in the same output file.

**NOTE:** The summary and log files will look identical to those seen in Section V.A but will contain more files.

## C. Allowing one error in the barcode

Under default settings, the script does not allow for any errors in the barcode. Sequences identified with one or more errors will be put in the "UNDETERMINED" file. The eight barcodes were designed in a manner that allows for multiple errors in the barcode, such that it is still possible to distinguish one barcode from any other barcodes in the set. Adding the qualifier `-allow` or `-a` to the end of either command for single paired files or directories will allow the script to identify the barcode even if one error is present. This command will increase the number of reads that are called with each individual barcode and decrease the number of undetermined reads (view our technical note for more details). The summary file (example above) will include information regarding how many mismatches were permitted.

- Example when using a single pair of files:

```
java -jar /path/to/TCR_Demux.jar /path/to/Read2 /path/to/Read1 -a
```

- Example when using a directory:

```
java -jar /path/to/TCR_Demux.jar /path/to/directory -a
```

## Contact us today

Please send your inquiries to our Technical Support department
by calling 1.800.662.2566 or emailing techUS@takarabio.com.

**Takara Bio USA, Inc.**
United States/Canada: +1.800.662.2566 • Asia Pacific: +1.650.919.7300 • Europe: +33.(0)1.3904.6880 • Japan: +81.(0)77.565.6999

**For Research Use Only. Not for use in diagnostic purposes.**