

Takara Bio USA, Inc.

Cogent™ NGS Immune Profiler v1.5 User Manual

Cat. Nos. 634466, 634467, 634480, 634481, 634776, 634777, & 634778
software v1.5
(111122)

Takara Bio USA, Inc.
2560 Orchard Parkway, San Jose, CA 95131, USA
U.S. Technical Support: technical_support@takarabio.com

United States/Canada	Asia Pacific	Europe	Japan
800.662.2566	+1.650.919.7300	+33.(0)1.3904.6880	+81.(0)77.565.6999

Page 1 of 32

Table of Contents

I.	Introduction	4
A.	Welcome to the Cogent NGS Immune Profiler Software.....	4
B.	What's New	4
II.	Before You Begin	4
A.	Supported Operating Systems.....	4
B.	Hardware Requirements	5
C.	Additional Software Dependencies	5
D.	User Account Requirements	5
E.	Required Input Files	5
III.	Software Overview.....	7
IV.	Installation and Configuration Requirements.....	8
A.	Uninstall Previous Instances of Immune Profiler	8
B.	Immune Profiler Download and Installation	8
C.	Verify Java Installation.....	8
D.	Python Installation and Verification	9
E.	Conduct Test Run with the Mini Datasets	11
F.	Uninstalling Immune Profiler	12
V.	Using Immune Profiler.....	12
A.	Best Practices.....	12
B.	Using the GUI (MacOS only).....	13
C.	From the Command Line (Linux or Mac)	17
D.	Processing Time	21
VI.	Immune Profiler Output	21
A.	Overview	22
B.	report and airr_report Folders	22
VII.	Cogent NGS Immune Viewer	26
VIII.	References	26
	Appendix A. Overview of Mini Dataset Sample Files	26
A.	Sample Input Data	27
B.	Sample Results Data	28
	Appendix B. Log Files	29
	Appendix C. More Output Details from Immune Profiler	30
A.	Preprocess Folder	30
B.	run_migec Folder.....	30
C.	run_mixcr Folder	31

Table of Figures

Figure 1. Cogent NGS Immune Profiler analysis workflow.....	7
Figure 2. Visual diagram of the <code>immune_profiler</code> directory, including files and folders.	8
Figure 3. How to verify the Java version of your OS	9
Figure 4. The Python 3.6 installation pop-up on MacOS.	10
Figure 5. How to verify the Python version of your OS	10
Figure 6. Running the required Python module check script and finding a module not installed.	10
Figure 7. Output to a successful check of required python modules.	11
Figure 8. Immune Profiler GUI.....	13
Figure 9. Immune Profiler GUI progress window at the completion of a successful analysis.	16
Figure 10. Output for the <code>python3 immune_profiler.py -h</code> command.	17
Figure 11. Folder structure and files found in <code>test_input/</code>	27
Figure 12. Folder structure and files found in <code>test_output/human/</code>	28

Table of Tables

Table 1. Example contents of a metadata CSV file, as would be viewed in a spreadsheet program	6
Table 2. Immune profiling kit prefix reference, mini dataset data directory, and mini dataset metadata file names.	11
Table 3. Mini dataset data results directory correlations to the immune profiling kits.....	12
Table 4. UI parameters for the mini dataset samples to match mini dataset sample output	15
Table 5. CLI parameters for the mini dataset samples to match mini dataset sample output	20
Table 6. Dataset parameters used for benchmark testing.....	21
Table 7. Machine specifications used for benchmark testing	21
Table 8. Benchmark results, per machine for each dataset	21
Table 9. Mapping statistics table files created based on the optional target region configuration.....	23
Table 10. <code><output name>_mig_[cdr3 fl]_mapping_stats.csv</code> column names and their descriptions.	23
Table 11. <code>report/<output name>_<sampleID>_mig_[cdr3 fl]_report.xlsx</code> column names and descriptions.	24
Table 12. <code>airr_report/<output name>_<sampleID>_mig_[cdr3 fl]_report.csv</code> column names and descriptions.	25
Table 13. Immune Profiler log files	29

I. Introduction

A. Welcome to the Cogent NGS Immune Profiler Software

Cogent NGS Immune Profiler Software v1.5 (referred to as Immune Profiler or CogentIP in this guide) is designed to analyze sequence data stored in FASTQ files generated by Illumina® sequencing platforms from libraries prepared using Takara Bio's human repertoire immune profiling kits (refer to the software compatibility table on the [bioinformatics portal](#) page at [takarabio.com](#) for more details). The output from CogentIP can then be imported into the [Cogent NGS Immune Viewer](#) (Section VII) to visualize the sequence data, such as in chord diagrams.

Written in Python3, Immune Profiler can be launched from a command line interface (CLI); MacOS users also have the option of running it via a graphical user interface (GUI). Immune Profiler incorporates two third-party programs, MIGEC and MiXCR, packaged and included for use only with this software under an [end-user license agreement \(EULA\)](#), acceptance of which requires the Immune Profiler user to be bound by and to comply with the terms before downloading and using Profiler.

IMPORTANT: Checking the box next to the [EULA](#) acceptance statement before submitting the completed form constitutes accepting and legally binding the Immune Profiler user to the terms of the EULA.

We recommend new users to read through this document prior to starting. There is also a [quick start guide](#) available to download, which is a streamlined reference document for installation and usage of the software.

B. What's New

Unless otherwise noted, the current version of software contains all features included in previous versions.

- **Cogent NGS Immune Profiler v1.5**
 - Support for [SMART-Seq® Human BCR \(with UMIs\)](#)
 - A full additional set of output files compliant with Adaptive Immune Receptor Repertoire (AIRR) Community standards ([Section VI](#))
 - Single installation package—compatible with both Linux and MacOS

NOTE: Find release notes for prior versions of CogentIP in the table on the [Cogent NGS Immune Profiler product page](#).

II. Before You Begin

A. Supported Operating Systems

- Mac OS X: El Capitan (Version 10.11 and up)
- Linux: CentOS 6 or higher, Redhat 7.5 or higher

NOTE: If the library is sequenced with more than 25 million reads, use the Linux version

B. Hardware Requirements

- Memory: 16 GB RAM
- Free disk space: at least 100 GB available hard drive space

NOTE: Required free disk space depends on the aggregate size of the input FASTQ files and should be 4X the total size of the FASTQ files to guarantee completion. If the total size of your input FASTQ files is greater than 100 GB, then the larger value from that calculation is the amount of recommended free disk space.

See [Section V.D](#) for information on performance benchmark results.

C. Additional Software Dependencies

- Java 1.8 or higher
- Python 3.6 or higher

D. User Account Requirements

The account used to install Immune Profiler needs to have administrative privileges on the server or workstation where it will be installed, including read/write (R/W) permissions for the folder in which it will be located.

Once installed, regular user accounts can be used to run the Immune Profiler executable, but these accounts need to have R/W permissions for the folder where the source metadata CSV is located.

If you are uncertain if the account being used to install or use Immune Profiler meets these criteria, please consult with your local IT for additional assistance.

E. Required Input Files

Immune Profiler requires paired FASTQ and metadata files as input.

1. FASTQ files

The Profiler has been validated to use FASTQ files with up to 25 million total reads on MacOS with 16 GB RAM, equivalent to Illumina MiSeq® platform sequencing capability. For deeper sequencing, we recommend using the Linux version.

The input files, either compressed (*.fastq.gz) or decompressed (*.fastq) format, can be stored in any directory on the server or workstation as long as the folder is not private or has read-write user restrictions that would prevent the files from being accessed by Immune Profiler.

2. Metadata file

The metadata file is a comma-separated value (CSV) file with the following characteristics:

- The output folder of results from running Immune Profiler will be written to the same directory location as the metadata file.
- The metadata CSV file needs to be created by the user in any directory on the server where Immune Profiler is installed and holds user-defined sample names and the path information to the matching FASTQ file names.

- It should have a header consisting of the following three elements: `sampleID`, `read1_file_name`, and `read2_file_name`.
 - `sampleID` is a user-defined, unique identifier for each sample; it should be less than 20 characters in length and only contain alphanumeric characters or hyphens. During the analysis stage, Immune Profiler scans the metadata file to check for the following conditions:
 - 1) Duplicate `sampleIDs`
 - 2) Underscores in the `sampleID` name
 - 3) All `sampleIDs` are 20 characters or less in length
 - 4) Blank lines

If any match is found to Conditions 1–3, Immune Profiler will display an error message noting which condition failed and terminate analysis. Edit the metadata file to fix the issue and relaunch Profiler.

If a blank line is found in the metadata file (Condition 4), Immune Profiler will ignore the empty line, display a warning, and continue processing the rest of the samples.
 - The `read1_file_name` and `read2_file_name` values should match the FASTQ file names corresponding to the sample specified by `sampleID`; directory location information should not be included. Immune Profiler will check and make sure these specified files exist in the FASTQ directory (described above); if no file matching the name is found, an error message is displayed, prompting the user to double-check both the FASTQ directory and FASTQ file names.

An example of the metadata file contents is shown in Table 1.

Table 1. Example contents of a metadata CSV file (above), as would be viewed in a spreadsheet program

sampleID	read1_file_name	read2_file_name
S1	S1_R1.fastq.gz	S1_R2.fastq.gz
S2	S2_R1.fastq.gz	S2_R2.fastq.gz
S3	S3_R1.fastq.gz	S3_R2.fastq.gz

Additional examples can be viewed in the metadata CSV files of mini dataset samples included with the software. See [Appendix A](#) for more information about the mini dataset sample and result files.

III. Software Overview

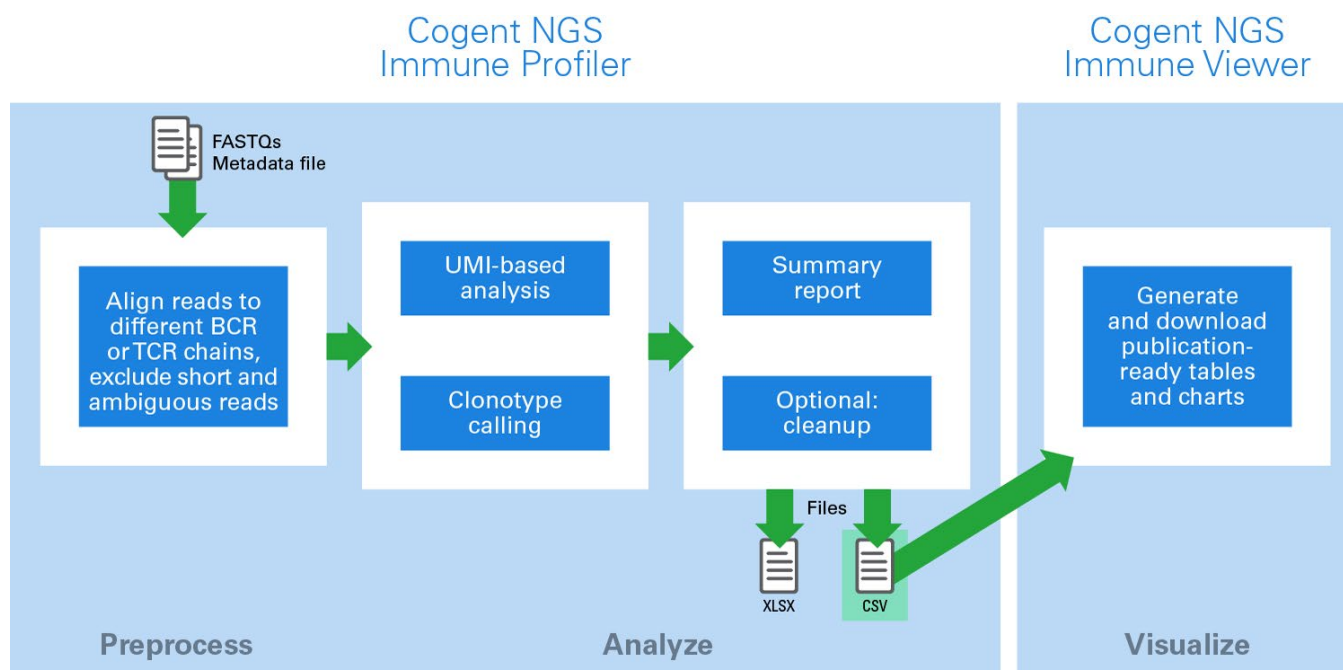


Figure 1. Cogent NGS Immune Profiler analysis workflow.

The CogentIP workflow consists of five steps and can be invoked using a graphical user interface (GUI) or can be run on the command line. Figure 1 illustrates at a high-level what the workflow consists of, while the list below expands on each step.

1. Preprocess:
 - Splits reads by matching read sequence to different receptor chains (allows one mismatch)
 - Excludes short reads (<30 bp) and reads ambiguously matched to multiple receptor chains
 - If linker-based correction is enabled, excludes read-failed correction
2. UMI-based analysis:
 - Groups reads into molecular identifier groups (MIG) using UMIs (Unique Molecular Identifiers)
 - Conducts sequencing error correction and exclude reads that fail abundance check
 - Generates collapsed-read FASTQs for downstream analysis
3. Clonotype calling:
 - Aligns reads to V(D)J sequences
 - Assembles alignments
 - Defines and reports clonotypes
4. Summary report:
 - Summarizes read QC statistics: chain-specific, short, undetermined, full-length chain (flc)
 - Summarizes mapping statistics: aligned, overlapped, mapped
 - Summarizes clonotype details: numbers, percentage, nucleotide, and amino acid sequence
 - Two output formats: file set identical to CogentIP v1.0, second set with AIRR-compliant header names
5. (Optional) Cleanup:
 - Configuration options available to user on a per run basis
 - Deletes intermediate files generated during processing (see [Section V.B](#) (GUI), [Section V.C](#) (CLI), and [Appendix C](#) for more information)

IV. Installation and Configuration Requirements

REMINDER: Administrative privileges on the server or workstation is required (Section II.D).

A. Uninstall Previous Instances of Immune Profiler

If an earlier version of Immune Profiler was installed on the server, it will need to be uninstalled prior to installing Cogent NGS Immune Profiler v1.5.

Follow the uninstall directions in [Section IV.F](#) ("Uninstalling Immune Profiler").

NOTE: If no version of Immune Profiler has ever been installed on the server, skip to the next section ([Section IV.B](#)).

B. Immune Profiler Download and Installation

Immune Profiler is available for download as a compressed file from takarabio.com/ngs-immune-profiler.

1. Download the installation ZIP file (Cogent_NGS_Immune_Profiler_Software_v1.5.zip), following the directions (a) on the page seen after submitting the sign-up form on the CogentIP product page or (b) in the confirmation email sent to the email address submitted in the form.
2. If this software is to be used by multiple users, put the package in a location anyone who will be using Immune Profiler can access and has computer permissions to use.
3. Decompress the software package in the directory it is to be installed in.

The following files should be included in the resulting `immune_profiler/` directory:

- Cogent NGS Immune Profiler v1.5 Quick Start Guide.pdf.
- Cogent NGS Immune Profiler v1.5 User Manual.pdf.
- ImmuneProfiler : the GUI launcher (for use on MacOS only).
- `immune_profiler.py` : main analysis script (Mac and Linux) .
- `required_python_module_check.py`: a script for to check for required python modules.
- `src/`: folder storing dependencies required by Immune Profiler.
- `test/` : folder contains a directory of test dataset files (`test_input/human/`) and a directory of example outputs generated by the test dataset files (`test_output/human/`). More information about this folder can be found in [Appendix A](#).

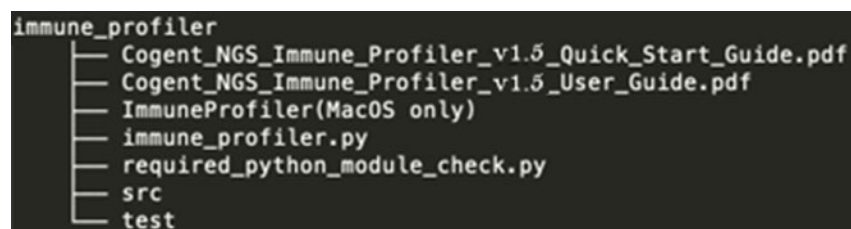


Figure 2. Visual diagram of the `immune_profiler` directory, including files and folders.

C. Verify Java Installation

1. Open a terminal window on the computer on which Immune Profiler will be installed:
 - a. **Mac:** The Terminal application is typically found under **Applications > Utilities > Terminal**. Alternatively, search for `terminal` in Spotlight search.

- b. **Linux:** If using Linux with a GUI shell, 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.

If using Linux on the command line interface (CLI), the CLI is the terminal window.

NOTE: You will need to use this terminal window for the next two sections; it is recommended not to close it until directed.

2. Verify the version of Java installed by typing:

```
java -version
```

into the terminal window. Text similar to Figure 3 should display.

```
$ java -version
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-b10)
OpenJDK 64-Bit Server VM (build 25.171-b10, mixed mode)
```

Figure 3. How to verify the Java version of your OS. After typing `java -version` into your terminal, you should see 'java version' and a number displayed in double-quotes. Verify that the first two number values are 1.8 or higher.

For assistance installing Java, visit the website:

https://java.com/en/download/help/download_options.xml

D. Python Installation and Verification

If Python is already installed on the system, skip to [Verifying Python Installation and Version](#).

1. Download and Install Python

1. Download Python from the website:

<https://www.python.org/downloads/>

Select the correct installation package for the operating system on which it will be installed. It should be version 3.6.0 or higher.

2. Install Python.

Mac users: accept the default settings.

(continued on next page)



Figure 4. The Python 3.6 installation pop-up on MacOS.

For additional assistance installing Python, visit the website:

<https://www.python.org/about/gettingstarted/>

2. Verifying Python installation and Version

In the same terminal window used to verify the version of Java, verify the version of Python installed by typing:

```
python3 -V or python3 --version
```

```
$ python3 -V
Python 3.6.1
```

Figure 5. How to verify the Python version of your OS. After typing `python3 -V` into your terminal, you should see a version number displayed. Verify that it is 3.6.0 or higher.

3. Verifying Installed Python Modules

In addition to the default modules installed by Python3, Immune Profiler also requires the module 'openpyxl' to be installed.

1. In the same terminal window used to verify the version of Java and Python, change to the directory where Immune Profiler was installed (`immune_profiler/`).
2. Type the command:

```
python3 required_python_module_check.py
```

If any required modules are missing, the script will list them on the terminal, as in Figure 6.

```
$ python3 required_python_module_check.py

Checkin for python packages required by Immune Profiler...

-- openpyxl is not installed.

See http://docs.python.org/3/installing/ for help with package installation.
```

Figure 6. Running the required Python module check script and finding a module not installed.

Otherwise, it will return, "All required packages are installed." (Figure 7).

```
Checking for python packages required by Immune Profiler...
All required packages are installed.
```

Figure 7. Output to a successful check of required python modules.

- If any packages are reported as missing by the check, install each of them (individually) with the following command typed into the terminal window:

```
python3 -m pip install <package_name>
```

or

```
pip3 install <package_name>
```

where <package_name> is replaced by the name of the package. E.g.,

```
python3 -m pip install openpyxl
```

or

```
pip3 install openpyxl
```

For additional assistance installing these modules, please refer to the website:

<https://docs.python.org/3/installing/>

- If you will be using the MacOS GUI version of Immune Profiler, the terminal window can be closed at this point. If using the CLI, keep it open and proceed to the next section.

E. Conduct Test Run with the Mini Datasets

After installation, an analysis should be done to verify the install using one of the the mini dataset sample files provided in `test/test_input/human/`. See [Appendix A](#) for more information about the sample mini dataset files. Table 2 can be used as an aid to determine which mini dataset may be of most interest to you.

NOTE: The five-character prefix value (abbreviation) in Table 2 will be used throughout the manual as shorthand for the output of the listed reagent kits, in terms of options within the software.

Table 2. Immune profiling kit prefix reference, mini dataset data directory, and mini dataset metadata file names.

Reagent kit	Prefix	Mini dataset data directory	Mini dataset metadata file
SMART-Seq Human BCR (with UMIs)	BCRv2	BCRv2_mini/	bcrv2_mini_meta.csv
SMARTer® Human BCR IgG IgM H/K/L Profiling Kit	BCRv1	BCRv1_mini/	bcrv1_mini_meta.csv
SMART-Seq Human TCR (with UMIs)	TCRv2	TCRv2_mini/	tcrv2_mini_meta.csv
SMARTer Human TCR a/b Profiling Kit v2			

- Select one of the mini datasets to test with (e.g., SMART-Seq Human BCR (with UMIs)). You will then use the FASTQ files in the mini dataset directory (e.g., BCRv2_mini/) and the mini dataset metadata file (e.g., bcrv2_mini_meta.csv) corresponding to your selected reagent kit.
- Follow the steps in [Section V.B](#) (UI) or [Section V.C](#) (CLI) to setup and execute an analysis run on your system.
- Compare the analysis results (1) generated by your run to the results provided in the `report/` subfolder ([Section VI.B](#)) to (2) the mini dataset output in `test/test_output/human/`.

Table 3 can be used to determine which mini dataset results directory with which to compare your output. Refer to [Appendix A](#) for more information about the mini dataset results data.

Table 3. Mini dataset data results directory correlations to the immune profiling kits.

Reagent kit	Prefix	Mini dataset results directory
SMART-Seq Human BCR (with UMIs)	BCRv2	BCRv2_mini_results/
SMARTer Human BCR IgG IgM H/K/L Profiling Kit	BCRv1	BCRv1_mini_results/
SMART-Seq Human TCR (with UMIs)	TCRv2	TCRv2_mini_results/
SMARTer Human TCR a/b Profiling Kit v2		

The installation is considered to be successful if the output to your test run matches the results stored in test/test_output/human/<kit-folder>/, where <kit-folder> is one of the mini dataset result directories listed in Table 3 relating to the dataset data you ran as a test (e.g., test/test_output/human/BCRv2_mini_results/).

F. Uninstalling Immune Profiler

1. Move any output files that you want to keep that are located in the immune_profiler/ directory to another location outside that folder.
2. Delete the immune_profiler/ folder, all its subfolders, and the files contained in it.

NOTE: If an older version of Immune Profiler was uninstalled in order to upgrade to a newer version, return to [Section IV.B](#) to continue the installation.

V. Using Immune Profiler

A. Best Practices

- As a reminder, after the initial installation and before analyzing your own data the first time, conduct a test run using the provided sample datasets and compare your output to the sample results ([Section IV.E](#)).
- The computer Immune Profiler is installed on should be plugged in and not running on battery (if a laptop) when a run is initiated. Since the profiling process may take some time to complete, depending on the size of the dataset being analyzed (see [Section V.D](#)), this recommendation is to prevent the computer from shutting down before the analysis is finished.
- The metadata and FASTQ files should be stored locally (on the same computer) to the Immune Profiler installation. Immune Profiler is designed to run on a single machine, and the computing speed and stability of the program degrade if analyzing data stored in a remote network location, even if on a mapped drive.
- For MacOS, the recommended upper input data limit is 25 million total reads (MiSeq-size data) due to processing constraints. If you have more data than this—e.g., ten FASTQ pairs, each with 5 million reads (50 million reads in total)—you can create two metadata files ([Section II.E](#)), with five samples in each, and launch two analysis runs sequentially to process all the data.

Once the first run is complete, check the free disk space to make sure there is enough for the second run ([Section II.B](#)) before launching it.

B. Using the GUI (MacOS only)

1. Procedure

NOTE: The examples used in this section to walk through the analysis procedure include file names corresponding to the mini dataset sample files included with the Immune Profiler package, detailed in [Appendix A](#).

1. Double-click on the executable file ImmuneProfiler located in the immune_profiler/ folder. This will launch the user interface (Figure 8). Depending on the dimension of your screen, there may be a scroll bar on the right side of the interface window.

Cogent™ NGS Immune Profiler Software
Version 1.5

Required information

FASTQ directory
Specify the folder containing the FASTQs to be analyzed
[Text Field] **Browse**

Metadata file
Specify the metadata file (csv) to use
[Text Field] **Browse**

Output name
Unique identifier for the analysis output files and the folder storing them
Note: less than 20 alphanumeric characters and/or hyphens
Example: test-Run123
[Text Field]

Specify receptor kit used
Choose either TCRv2, BCRv1 or BCRv2
[Select Option]

Target
Specify target regions where reads should map to:
[Select Option]

Optional configuration

UMI cutoff
Specify an integer to use as the UMI cutoff
[Text Field]

Keep intermediate
☐ Yes

Perform linker-based correction?
☐ Yes

Cancel **Start**

Figure 8. Immune Profiler GUI.

2. Populate the required information fields.

- a. FASTQ directory: use the [Browse] button to locate and specify the FASTQ directory for the Immune Profiler.

Example:

```
/immune_profiler/test/test_input/human/BCRv2_mini/
```

- b. Metadata file: use [Browse] to locate and select the metadata .csv file associated with the FASTQ files in the directory from 2a. When doing the test run, refer to [Table 2](#) for help determining which metadata file is of most interest to you.

Example:

```
/immune_profiler/test/test_input/human/bcrv2_mini_meta.csv
```

- c. Choose an "Output name" for this analysis run. This is an alphanumeric string, not a file path. The output name string is used to name the output folder created by the analysis and as a prefix for all the results files. Note that:

- The output folder will be created in the same directory location as the metadata CSV file ([Section II.E](#)). User accounts used to run Immune Profiler, therefore, must have read/write permissions to the folder so the output folder can be created ([Section II.D](#)).
- The output name should be less than 20 characters in length and only contain alphanumeric characters and/or hyphens.
- The output name should be different from the name of the parent folder of the metadata file.
- Immune Profiler will check if:
 - 1) A folder matching the output name string already exists in the metadata file location.
 - 2) The output name is identical to the parent folder name.

If a match is found to either case, Immune Profiler will terminate the analysis with the corresponding error message:

- 1) Analysis dir already exists: <folder name>
- 2) The output folder name you defined is identical to its upper folder, please rename.

- d. "Specify receptor type to analyze" using the dropdown menu, choose 'BCRv2', 'BCRv1', or 'TCRv2'. Refer to [Table 2](#), if needed, to correlate these abbreviations with a particular kit type.

- e. Specify "Target region" using the dropdown menu: 'CDR3', 'Full_length', or 'Both'

This option allows a user to specify analysis on the CDR3 region only, the full-length transcript, or to analyze both CDR3 and full-length sequences in a single run (Both).

- For BCR, the read will cover the full-length of the V(D)J sequence.
- For TCR, it will depend on the actual read length; please refer to the [SMART-Seq Human TCR \(with UMIs\) User Manual](#), Appendix B ("Guidelines for Library Sequencing and Data Analysis") for more information about this concept.

NOTE: If you are interested in both CDR3 and full-length results, select Both. This will decrease the total processing time compared to launching two independent runs.

3. (Optional) Make selections in the optional configuration fields.
 - a. "Keep intermediate files?": default = unchecked (false)

During analysis, Immune Profiler will create interim files to hold data temporarily until used to generate the final output. These intermediate files include preprocessing FASTQs and binary-form alignment and assembling data files.

By default, Immune Profiler deletes these files once analysis steps associated with them have been completed as they are large and may quickly consume available disk space. By checking this option, the intermediate files will be retained.
 - b. "Perform linker-based correction?": default = unchecked (false)

PCR errors, sequencing errors, or deletion or insertion of one or more nucleotides could cause a frameshift of final read sequences. To benchmark and conduct quality control on these kinds of errors, Immune Profiler offers linker-based correction, which compares the read sequence in certain regions on the read with the designed linker sequence (Arguel 2017; Turchaninova 2016; Vander Heiden 2014).

By default, this linker-based correction is not performed. When the option is selected, this check is performed and if a frameshift is identified in a read, it is removed from downstream analysis.
4. Once all desired parameters are populated, click [Start] to begin the analysis.

2. Example

To generate results identical to the ones in `test/test_output/human/`, choose the parameters for the desired receptor type from the appropriate column in Table 4:

Table 4. UI parameters for the mini dataset samples to match mini dataset sample output

Parameters	BCRv2	BCRv1	TCRv2
FASTQ directory	%PROFILER_HOME%/test/test_input/human/BCRv2_mini	%PROFILER_HOME%/test/test_input/human/BCRv1_mini	%PROFILER_HOME%/test/test_input/human/TCRv2_mini
Metadata file	%PROFILER_HOME%/test/test_input/human/bcrv2_mini_meta.csv	%PROFILER_HOME%/test/test_input/human/bcrv1_mini_meta.csv	%PROFILER_HOME%/test/test_input/human/tcrv2_mini_meta.csv
Output name	test-run	test-run	test-run
Specify receptor type to analyze	BCRv2	BCRv1	TCRv2
Target region	CDR3	CDR3	CDR3
Keep intermediate files?	Unchecked	Unchecked	Unchecked
Perform linker-based corrections?	Unchecked	Unchecked	Unchecked

%PROFILER_HOME% is an abbreviation for the full directory path where the Immune Profiler software is installed.

E.g., if Immune Profiler is installed in `/home/user/bin`, then:

%PROFILER_HOME% = `/home/user/bin/immune_profiler`

1. The output string (folder name and file prefix) will be:

```
test-run
```

 and the output folder will be created in:

```
/immune_profiler/test/test_input/human/
```

 as:

```
/immune_profiler/test/test_input/human/test-run/
```
2. "Specify the receptor type" to analyze using the dropdown menu
3. Specify the "Target region" using the dropdown menu: 'CDR3'
4. Keep the defaults for the optional configuration fields (unchecked)
5. Once all parameters are populated, click [Start] to begin the analysis.
6. An analysis progress window will replace the Immune Profiler GUI (Figure 9, background).
7. Once the analysis is finished, a pop-up window will display over the progress window with the message "Execution finished / Program completed successfully!" (Figure 9, foreground)

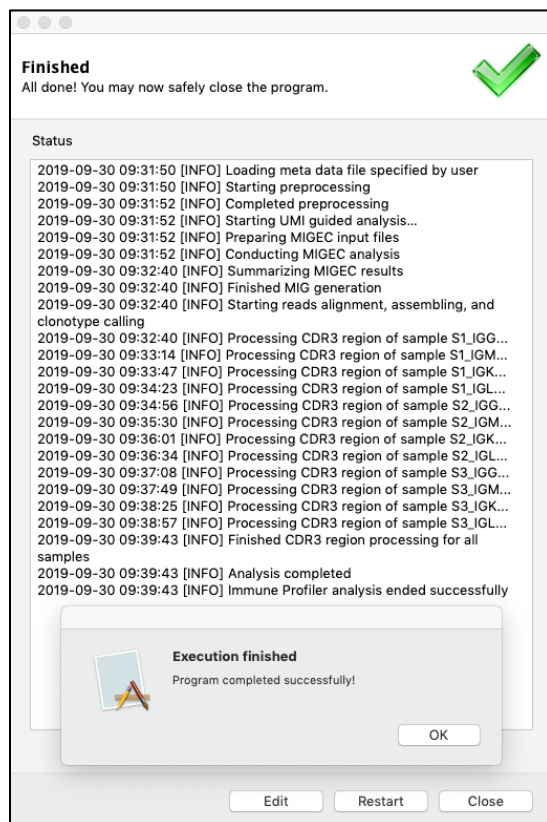


Figure 9. Immune Profiler GUI progress window at the completion of a successful analysis.

C. From the Command Line (Linux or Mac)

1. CLI Overview

With Linux or as an alternate method on Mac, Immune Profiler is launched via command line interface (CLI) utilizing the `immune_profiler.py` script. This script can be launched either from within the `immune_profiler/` directory or from any location (working directory) on the Linux server where Immune Profiler software is installed if the full path to the script is specified.

E.g., if `%PROFILER_HOME% = /home/user/bin/immune_profiler`, the script can be called with:

```
/home/user/bin/immune_profiler/immune_profiler.py
```

The full list of arguments can be accessed with the `-h` option (Figure 10):

- From within the `immune_profiler/` directory:

```
python3 immune_profiler.py -h
```

- From any location on the server:

```
python3 %PROFILER_HOME%/immune_profiler.py -h
```

```
$ python3 immune_profiler.py -h
usage: immune_profiler.py [-h] -r {TCRv2,BCRv1,BCRv2} -f FASTQ_DIR -m META_FILE
                        -o OUT_NAME -t {CDR3,FULL_length,Both} [-k] [-l] [-u]

Immune_profiler.py: A script to analyze sequence data stored in fastq files and
generated from Takara Human TCR/BCR kit (with UMI). User options are designed
to simplify analysis procedure derived from Takara protocols.

Required arguments:
  -r {TCRv2,BCRv1,BCRV2}, --receptor_type {TCRv2,BCRv1,BCRV2}
                        specify receptor type: TCRv2 or BCRv1 or BCRv2
  -r FASTQ_DIR, --fastq_dir FASTQ_DIR
                        A folder stores all input FASTQs
  -m META_FILE, --meta_file META_FILE
                        A file contains sample ID and corresponding FASTQ pair
  -o OUT_NAME, --output_name OUT_NAME
                        Name an output directory to be created to store results
                        and use as file prefix; directory name should be less
                        than 20 characters
  -t {CDR3,Full_length,Both}, --target_region {CDR3,Full_length,Both}
                        specify target regions reads should map to

Optional arguments:
  -h, --help            show this help message and exit
  -k, --keep_inter_file
                        decide if keep intermediate files, including
                        MIXCR files & preprocessed FASTQs [Default: False]
  -l, --linker_correction
                        decide if remove reads based on sequence match of
                        linker [Default: False]
  -u, --umi_cutoff      specify an integer to use as the UMI cutoff
                        [Default: '']
```

Figure 10. Output for the `python3 immune_profiler.py -h` command.

There are two types of arguments: required information and optional configurations. Users must specify all required arguments to launch Immune Profiler. The optional arguments have default values and can be omitted or included based on the analysis needs.

2. Required information arguments

See the example commands below ("Command Line Examples") for how each of these arguments might be configured.

- `-r` : the receptor type of the data files to be analyzed: BCRv1, BCRv2, or TCRv2

Example:

If the input files represent data for samples from The SMART-Seq Human TCR (with UMIs) or SMARTer Human TCR a/b Profiling Kit v2, then this argument and parameter would be typed:

```
-r TCRv2
```

- `-f` : path to the FASTQ folder

If there is an error message regarding files, check either the FASTQ folder path or metadata file for typos.

NOTE: Only FASTQs listed in the metadata file are processed for analysis, even if additional FASTQ files are stored in the specified FASTQ folder.

- `-m` : the path to and name of the metadata file (described in [Section II.E](#))

REMINDER: FASTQ files configured in the metadata file should exactly match the name of the FASTQ file in the directory above. Immune Profiler is case-sensitive.

- `-o` : output name

This is an alphanumeric string, not a file path. The output name string is used to name the output folder created by the analysis and as a prefix for all the results files. Note that:

- The output folder will be created in the same directory location as the metadata CSV file ([Section II.E](#)). User accounts used to run Immune Profiler, therefore, must have read/write permissions to the folder so the output folder can be created ([Section II.D](#)).
- The output name should be less than 20 characters in length and only contain alphanumeric characters and/or hyphens
- The output name should be different from the name of the parent folder of the metadata file
- Immune Profiler will check if:
 - 1) A folder matching the output name string already exists in the metadata file location
 - 2) The output name is identical to the parent folder name.

If a match is found to either case, Immune Profiler will terminate the analysis with the corresponding error message:

- 1) Analysis dir already exists: <folder name>
- 2) The output folder name you defined is identical to its upper folder, please rename.

- `-t` : specify target regions reads should map to: CDR3, Full_length, or Both

This option allows a user to specify analysis on only the CDR3 region, the full-length transcript, or to analyze both CDR3 and full-length sequences in a single run (Both). For BCR, the read will cover the full-length of the V(D)J sequence. For TCR, it will depend on the actual read length; please refer to the [SMART-Seq Human TCR \(with UMIs\) User Manual](#), Appendix B ("Guidelines for Library Sequencing and Data Analysis") for more information about this concept.

NOTE: If you are interested in both CDR3 and full-length results, select `Both`. This will decrease the total processing time compared to launching two independent runs.

Example:

If you want to do a full-length analysis, then this argument and parameter would be typed:

```
-t Full_length
```

3. Optional configuration arguments

- `-k` : condition to keep intermediate files

(default: false)

During the analysis, Immune Profiler will create interim files to hold data temporarily until used to generate the final output. These intermediate files include preprocessing FASTQs and binary-form alignment and assembling data files.

By default, Immune Profiler deletes these files once analysis steps associated with them have been completed as they are large and may quickly consume available disk space. By checking this option, the intermediate files will be retained.

- `-l` : condition to perform linker-based correction

(default: false)

PCR errors, sequencing errors, or deletion or insertion of one or more nucleotides could cause a frameshift of final read sequences. To benchmark and conduct quality control on these kinds of errors, Immune Profiler offers linker-based correction, which compares the read sequence in certain regions on the read with the designed linker sequence (Arguel 2017; Turchaninova 2016; Vander Heiden 2014).

By default, this linker-based correction is not performed. When the option is selected, this check is performed and if a frameshift is identified in a read, it is removed from downstream analysis.

- `-u` : specify an integer to use as umi_cutoff.

(default: 1)

'1' is the minimum value for this argument. To increase the confidence of the analysis result, the argument value can be increased, but we recommend a maximum value of '3'. Higher umi_cutoff values may result in insufficient reads per UMI to obtain meaningful data.

4. Command line examples

To generate results identical to the ones in test/test_output/, choose the parameters for the desired receptor type from the appropriate column in Table 5:

Table 5. CLI parameters for the mini dataset samples to match mini dataset sample output

Parameters	BCRv2	BCRv1	TCRv2
FASTQ directory	PROFILER_HOME%/test/test_input/human/BCRv2_mini	PROFILER_HOME%/test/test_input/human/BCRv1_mini	PROFILER_HOME%/test/test_input/human/TCRv2_mini
Metadata file	PROFILER_HOME%/test/test_input/human/bcrv2_mini_meta.csv	PROFILER_HOME%/test/test_input/human/bcrv1_mini_meta.csv	PROFILER_HOME%/test/test_input/human/tcrv2_mini_meta.csv
Output name	BCRv2	BCRv1	TCRv2
Specify receptor type to analyze	BCRv2	BCRv1	TCRv2
Target region	CDR3	CDR3	CDR3
Keep intermediate files?	unchecked	unchecked	unchecked
Perform linker-based corrections?	unchecked	unchecked	unchecked

%PROFILER_HOME% is an abbreviation for the full directory path where the Immune Profiler software is installed.

E.g., if Immune Profiler is installed in /home/user/bin, then:

```
%PROFILER_HOME% = /home/user/bin/immune_profiler
```

NOTE: The following commands should be typed all at one CLI prompt.

- The first example command will analyze the BCRv2 mini dataset to generate identical report files to /test/test_output/human/BCRv2_mini_results.

```
$ python3 immune_profiler.py -r BCRv2
-f %PROFILER_HOME%/test/test_input/human/BCRv2_mini
-m %PROFILER_HOME%/test/test_input/human/bcrv2_mini_meta.csv
-o BCRv2
-t CDR3
```

- This second example will analyze the TCRv2 sample dataset to generate identical report files to /test/test_output/TCRv2_mini_results

```
$ python3 immune_profiler.py -r TCRv2
-f %PROFILER_HOME%/test/test_input/human/TCRv2_mini
-m %PROFILER_HOME%/test/test_input/human/tcrv2_mini_meta.csv
-o TCRv2
-t CDR3
```

D. Processing Time

The run time of the pipeline will vary widely based on the specifications of the computer or server on which it is run. The information in this section is provided for comparison purposes to extrapolate for your own system.

1. Test parameters

The following files specifications were used to test a small and large dataset:

Table 6. Dataset parameters used for benchmark testing

Dataset parameters	Dataset 1 (small)	Dataset 2 (large)
Number of FASTQ files	4	2
Reads per file	~2.6 million	~13.4 million
Aggregate FASTQ file size (GB)	4.1	11.4

The following two machines were used to test both datasets:

Table 7. Machine specifications used for benchmark testing. Both machines were 64-bit (x86_64).

Hardware specification	Linux 1	MacOS X
Operating System	CentOS 6.10	MacOS Mojave 10.14.6
CPU	32-Core Intel Xeon CPU E7-8837 @ 2.67 GHz	Intel Core i5 @ 2.7 GHz
Memory (RAM)	128 GB	16 GB

2. Test Results

The following benchmark data was generated on the two machines:

Table 8. Benchmark results, per machine for each dataset

Dataset	Total runtime	
	Linux 1	MacOS X
Dataset 1 (small)	27 min	47 min
Dataset 2 (large)	2 hr 50 min	4 hr 5min

VI. Immune Profiler Output

NOTE: In this section, the following shortcut references are used:

- %PROFILER_HOME% is an abbreviation for the full directory path where the Immune Profiler software is installed.

E.g., if Immune Profiler is installed in /home/user/bin, then:

```
%PROFILER_HOME% = /home/user/bin/immune_profiler
```

- <sampleID> is a generic phrase that represents the value of the corresponding sample ID as defined in the metadata file (see [Section II.E](#)).

A. Overview

The output files and folder structure depend on the optional configuration arguments selected when running the tool. The potential folders found in the output folder include:

- `report/` : This folder summarizes major statistics collected by the previous workflow steps and merges information into files that can be viewed in a spreadsheet program (CSV and MS-Excel formats). This folder is covered in greater detail in [Section VI.B](#) (below).
- `airr_report/`: The contents of the folder are same as the folder `report/`. Files inside this folder use AIRR Community standard column names.
- `preprocess/` : contains intermediate FASTQ files created during preprocessing. If the "Keep intermediate files?" option is set to the default (not checked) in the configuration, this folder will be deleted prior to analysis completion.
- `run_migec/` : stores files generated during UMI-based read correction and collapsed-read FASTQs.
- `run_mixcr/` : stores files for read alignment, assembling, and clonotype calling during processing.

For more information about the `preprocess/`, `run_migec/`, and `run_mixcr/` folders, please refer to [Appendix C](#).

B. report and airr_report Folders

Immune Profiler summarizes major statistics collected by the workflow steps (Section V) and merges the results into comma-separated value (CSV) files and MS-Excel XLSX files that can be viewed in a spreadsheet program. These files are written to and can be found in the `report/` and `airr_report/` folders.

There are slight differences between the files in the two folders, but many have the same or similar names and they contain identical information. The two primary differences between the two sets of output files in the folders are the following:

1. All files in the `airr_report/` folder use column names compliant with AIRR community standards. For more information about this, please refer to: https://docs.airr-community.org/en/stable/swtools/airr_swtools_standard.html
2. In keeping with AIRR standards, all files in the `airr_report/` folder are in CSV format. Sample-level report files in the `report/` folder are in XLSX format and have multiple sheets (tabs).

Example output files for each receptor type are included in the mini dataset sample output in `%PROFILER_HOME%/test/test_output/human/` (see [Appendix A](#)). These sample files follow the output format of the `report/` folder.

1. <output name>_sample_QC_stats.csv

The QC (quality control) statistics table. For each sample (row), Immune Profiler summarizes the number of reads and read % that are assigned to different chains*, chains shorter than 30 bp in length (short), from undetermined chain (undetermined), failed linker-based correction (flc), and total chains.

*The different chains, by kit type, are listed below:

- **BCRv1:** IgG, IgM, IgK, and IgL
- **BCRv2:** IgG, IgM, IgK, IgL, IgA, IgD, IgE
- **TCRv2:** TRA and TRB

2. `<output name>_mig_[cdr3|fl]_mapping_stats.csv`

The mapping statistics table for all samples that are analyzed in the same Immune Profiler run for the target region, i.e., CDR3 or Full_length, specified in the configuration. The string `[cdr3|fl]` means either CDR3 or Full_length. Table 9 lists which files will be seen depending on the target region option selected.

Table 9. Mapping statistics table files created based on the optional target region configuration

Target region	File name
Both (default)	<code><output name>_mig_cdr3_mapping_stats.csv</code> <code><output name>_mig_fl_mapping_stats.csv</code>
CDR3	<code><output name>_mig_cdr3_mapping_stats.csv</code>
Full_length	<code><output name>_mig_fl_mapping_stats.csv</code>

The output file contains mapping information and a clonotype summary for all combinations of samples and chains. The columns names for each folder (`report/` and `airr_report/`) and what information the columns represent are listed in Table 10.

Table 10. `<output name>_mig_[cdr3|fl]_mapping_stats.csv` column names and their descriptions.

Column name (<code>report/</code>)	Column name (<code>airr_report/</code>)	Description
—	<code>organism_id</code>	(<code>airr_report</code> only) This value will always be 'human'
—	<code>sample_processing_id</code>	(<code>airr_report</code> only) The value <code><sampleID></code>
sample type	<code>sample_type</code>	User-defined sampleID plus chain type, i.e., <code><sampleID>_<chain type></code> .
total reads	<code>total_read</code>	Total reads in original FASTQs of corresponding sample + chain type.
total MIG	<code>total_mig</code>	Total molecular identifier groups (MIG) classified by MIGEC after read check.
UMI threshold	<code>umi_threshold</code>	The UMI threshold used to discard rare reads with too few UMIs. Efficient error correction preferably requires more than five reads per UMI, and the minimum requirement is three (Shugay 2014; Turchaninova 2016).
number of reads after MIG collapse	<code>number_of_reads_after_mig_collapse</code>	After MIG collapse, reads belonging to the same MIG are collapsed into a single read. These reads are stored in FASTQs regenerated under <code>run_migec/assemble</code> ; this column reports total reads in these files.
aligned	<code>aligned</code>	Number of reads aligned to target region by MiXCR.
pair-read overlap	<code>pair-read_overlap</code>	Number of pair-reads that overlap with each other.
overlapped and aligned	<code>overlapped_and_aligned</code>	Number of reads that have overlap and also were aligned to the target region by MiXCR.
clonotype count	<code>clonotype_count</code>	The total number of the clonotype identified after MiXCR procedure.
<code><chain counts></code>	<code><chain counts></code>	Column names vary here: if BCR, count of IgG, IgM, IgK, IgL, IgA, IgD, IgE, and IgH (lack of a constant region) will be listed, or in all caps (IGG, IGM, etc.) for the AIRR format; If TCR, columns of TRA and TRB will be listed.

3. **report/<output name>_<sampleID>_mig_[cdr3|fl]_report.xlsx**

A sample-level report file summarizing mapping statistics of the particular sample specified by <sampleID> and clonotype details for each chain.

- The first tab, stats, contains sample-specific mapping statistics. It is a subset of the previous file (<output name>_mig_[cdr3|fl]_mapping_stats.csv) and identical to it in table structure.
- The rest of the tabs are clonotype details for each chain type, with the columns in Table 11.

Table 11. report/<output name>_<sampleID>_mig_[cdr3|fl]_report.xlsx column names and descriptions.

Column name	Description
Read Count*	Number of reads with which a particular clonotype is identified.
Fraction	Fraction of read count over total reads.
Clonal Sequence	User-defined sampleID plus chain type, i.e., <sampleID>_<chain type>. <ul style="list-style-type: none"> • If target region is CDR3 only, this sequence is the sequence in CDR3 region only. • If target region is Full_length, the corresponding clonotype sequence is reported here.
Clonal Sequence Quality	Sequence quality score†.
CDR3 Min Quality	Minimal quality score used as a threshold. Only sequence reads with their quality score above this threshold would be aligned and assembled.
CDR3 Sequence	CDR3 region sequence.
CDR3 Amino Acid Sequence	Amino Acid Sequence of CDR3 region.
Clonal type	Chain type categories. For BCR, the value could be: <ul style="list-style-type: none"> • BCRv2: IgG, IgH†, IgM, IgK, IgL, IgA, IgD, or IgE • BCRv1: IgG, IgH†, IgM, IgK, or IgL
Frame Shift	A mark indicates if any frameshift is found in a read.
Stop Codon	A mark indicates if any stop codon is found in a read.
Amino Acid Length	Total amino acid length for the corresponding clonotype.
[V D J C] segment	the most likely segment type for each of the four categories. There is a column for each type (i.e., V segment, D segment, etc.)
all [V D J C] hits	all possible V segment types for each of the four categories. There is a column for each type (i.e., all J hits).

*The read here is a collapsed read, mapped to the FASTQs in the run_migec/assemble/ folder. Immune Profiler assumes UMIs are on Read 1, while the reagent kit chemistries have UMIs on Read 2. The FASTQ file names, therefore, correspond to their opposite read.

†Since the collapsed read is used, the quality score is an artificial assignment by the software.

‡A clonal_type value of IgH means the sequence is a heavy chain region that doesn't contain a constant region that would identify the chain as either an IgG or IgM.

4. **airr_report/<output name>_<sampleID>_mig_[cdr3|fl]_report.csv**
A sample-level report file summarizing mapping statistics of the particular sample specified by <sampleID> and clonotype details for each chain.

Unlike the XLSX file in the report/ folder, this similarly named CSV file contains only one tab representing an aggregate of the clonotype details for each chain type, with the columns in Table 12.

Table 12. airr_report/<output name>_<sampleID>_mig_[cdr3|fl]_report.csv column names and descriptions.

Column name	Description
organism_id	Value will always be 'human'
sample_processing_id	Value of <sampleID>
read_count*	Number of reads with which a particular clonotype is identified.
fraction	Fraction of read count over total reads.
clonal_sequence	User-defined sampleID plus chain type, i.e., <sampleID>_<chain type>. <ul style="list-style-type: none"> If target region is CDR3 only, this sequence is the sequence in CDR3 region only. If target region is Full_length, the corresponding clonotype sequence is reported here.
clonal_sequence_quality	Sequence quality score†.
cdr3_min_quality	Minimal quality score used as a threshold. Only sequence reads with their quality score above this threshold would be aligned and assembled.
cdr3_sequence	CDR3 region sequence.
cdr3_amino_acid_sequence	Amino Acid Sequence of CDR3 region.
clonal_type	Chain type categories. For BCR, the value could be: <ul style="list-style-type: none"> BCRv2: IgG, IgH‡, IgM, IgK, IgL, IgA, IgD, or IgE BCRv1: IgG, IgH‡, IgM, IgK, or IgL
frame_shift	A mark indicates if any frameshift is found in a read.
stop_codon	A mark indicates if any stop codon is found in a read.
junction_length_aa	Total amino acid length for the corresponding clonotype.
[v d j c]_segment	The most likely segment type for each of the four categories. There is a column for each type (i.e., v_segment, d_segment, etc.)
all_[v d j c]_hits	All possible V segment types for each of the four categories. There is a column for each type (i.e., all_j_hits).

*The read here is a collapsed read, mapped to the FASTQs in the run_migec/assemble/ folder. Immune Profiler assumes UMIs are on Read 1, while the reagent kit chemistries have UMIs on Read 2. The FASTQ file names, therefore, correspond to their opposite read.

†Since the collapsed read is used, the quality score is an artificial assignment by the software.

‡A clonal_type value of IgH means the sequence is a heavy chain region that doesn't contain a constant region that would identify the chain as either an IgG or IgM.

5. <sampleID> folders

These subfolders are created for individual samples and contain CSV files that store clonotype details for each chain type. The files have the following naming convention:

```
<sampleID>_<chain-type>_mig_[cdr3|fl]_result.csv
```

The *_result.csv files are made available for use with Cogent NGS Immune Viewer (Section VII) or for advanced users for downstream analysis.

- In the report/ folder—each CSV file is identical to the sample-level clonotype tabs included in <output name>_<sampleID>_mig_[cdr3|fl]_report.xlsx
- In the airr_report/ folder—each CSV file is the clonotype-specific subset of <output name>_<sampleID>_mig_[cdr3|fl]_report.csv

VII. Cogent NGS Immune Viewer

Cogent NGS Immune Viewer (referred to as Immune Viewer) takes as input:

- **(Recommended)** *_result.csv files generated by Immune Profiler from the report/<sampleID> and airr_report/<sampleID> folders ([Section VI.B.5](#), above), -or-
- airr_report/<output name>_<sampleID>_mig_[cdr3|fl]_report.csv ([Section VI.B.4](#))

then creates visualizations (charts) or tabulated outputs and publication-ready plots for download.

For more information on the Immune Viewer, please visit the software website page at takarabio.com/ngs-immune-viewer or the [Cogent NGS Immune Viewer v1.0 User Manual](#).

VIII. References

- Arguel, MJ. *et al.* A cost effective 5' selective single cell transcriptome profiling approach with improved UMI design. *Nucleic Acids Res.* **45**, e48 (2017).
- Bolotin, D. A. *et al.* MiXCR: software for comprehensive adaptive immunity profiling. *Nat. Methods* **12**, 380–381 (2015).
- Shugay, M. *et al.* Towards error-free profiling of immune repertoires. *Nat. Methods* **11**, 653–655 (2014).
- Turchaninova, M. A. *et al.* High-quality full-length immunoglobulin profiling with unique molecular barcoding. *Nat. Protoc.* **11**, 1599–1616 (2016).
- Vander Heiden, J. A. *et al.* pRESTO: a toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires. *Bioinformatics* **30**, 1930–1932 (2014).

Appendix A. Overview of Mini Dataset Sample Files

Mini dataset sample files are provided in the Immune Profiler installation within the folder:

```
immune_profiler/test/test_input/human/
```

These should be used both to verify Immune Profiler is installed correctly ([Section IV.E](#)) and to familiarize yourself with the operative steps to use the software.

A. Sample Input Data

Three sets of sample data are provided in the `test/test_input/human/` directory. Refer to Figure 11 for a visual representation of the subfolder contents:

```
test_input/human/
├── bcrv1_mini_meta.csv
├── BCRv1_mini
│   ├── S1_R1.fastq.gz
│   ├── S1_R2.fastq.gz
│   ├── S2_R1.fastq.gz
│   ├── S2_R2.fastq.gz
│   ├── S3_R1.fastq.gz
│   └── S3_R2.fastq.gz
├── bcrv2_mini_meta.csv
├── BCRv2_mini
│   ├── S1_R1.fastq.gz
│   ├── S1_R2.fastq.gz
│   ├── S2_R1.fastq.gz
│   ├── S2_R2.fastq.gz
│   ├── S3_R1.fastq.gz
│   └── S3_R2.fastq.gz
├── tcrv2_mini_meta.csv
└── TCRv2_mini
    ├── S4_R1.fastq.gz
    ├── S4_R2.fastq.gz
    ├── S5_R1.fastq.gz
    ├── S5_R2.fastq.gz
    ├── S6_R1.fastq.gz
    └── S6_R2.fastq.gz
```

Figure 11. Folder structure and files found in `test_input/`.

1. Mini dataset BCRv1 input files include:
 - `bcrv1_mini_meta.csv`
 - `BCRv1_mini/` folder : Read1 and Read2 FASTQ files for three samples: S1, S2, and S3, each with 16,000 paired-end reads
2. Mini dataset BCRv2 input files include:
 - `bcrv2_mini_meta.csv`
 - `BCRv2_mini/` folder : Read1 and Read2 FASTQ files for three samples: S1, S2, and S3, each with 20,000 paired-end reads
3. Mini dataset TCRv2 input files include:
 - `tcrv2_mini_meta.csv`
 - `TCRv2_mini/` folder : Read1 and Read2 FASTQ files for three samples: S4, S5, and S6, each with 3,000, 2,500 and 2,000 paired-end reads

B. Sample Results Data

Three sets of sample result files are provided in the `test/test_output/human/` folder, corresponding to the three sets of input data in Section A. Refer to Figure 12 for a visual representation of the folder contents.

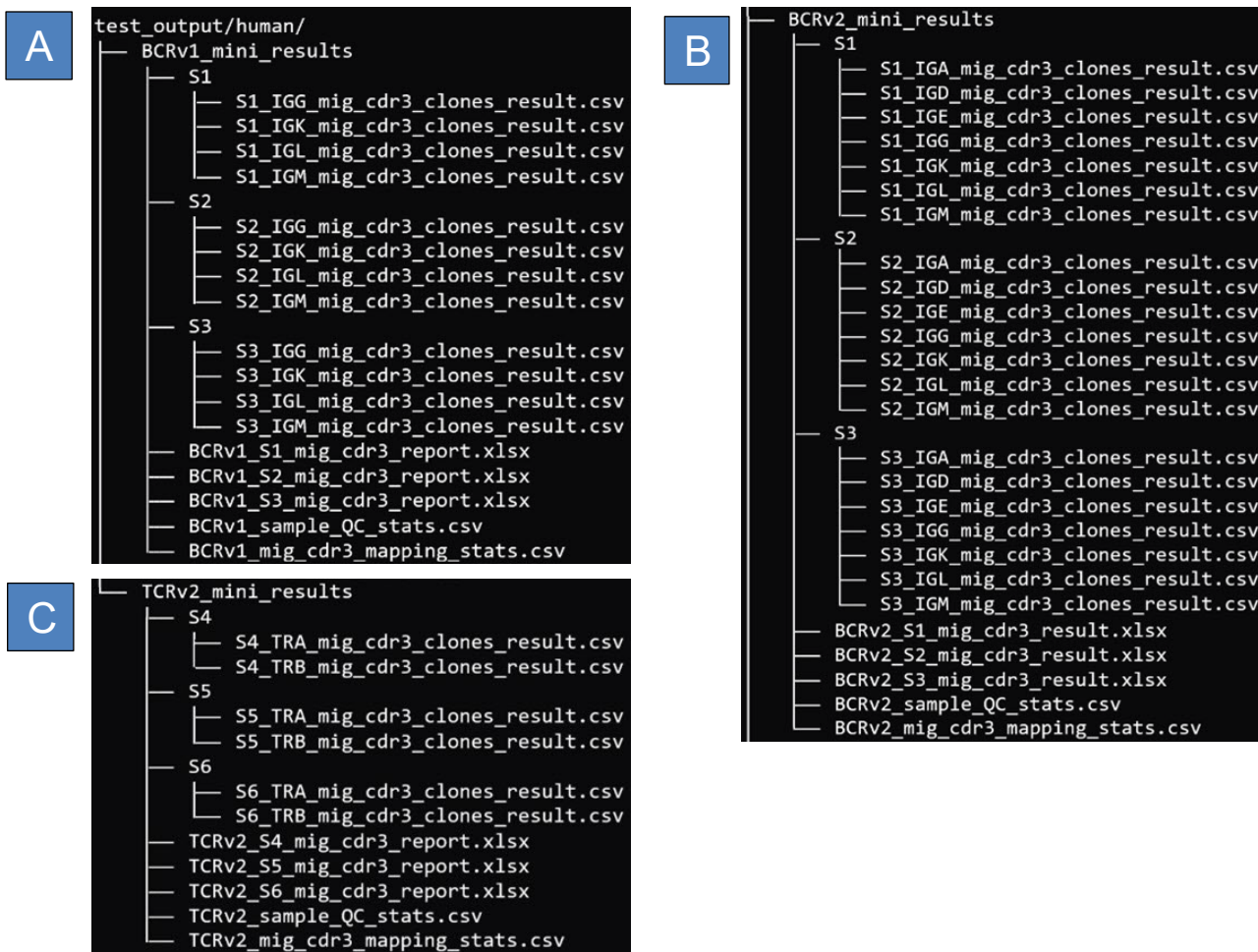


Figure 12. Folder structure and files found in `test_output/human/`. (Panel A) Mini dataset files for BCRv1. (Panel B) Mini dataset files for BCRv2. (Panel C) Mini dataset files for TCRv2

In the list below, `<RECEPTOR>` is the immune cell receptor type of the data being processed (options: BCRv1, BCRv2, -or- TCRv2)

- Files**

- `<RECEPTOR>_<sampleID>_mig_cdr3_report.xlsx`: summary statistics of individual samples, identified by `sampleID` (as defined in the metadata input file), together with clonotype details derived from all the receptor chains
 - `<RECEPTOR>_sample_QC_stats.csv`: quality control statistics for all samples of the receptor type (i.e, `BCRv2_sample_QC_stats.csv` are QC stats for all BCRv2 samples)
 - `<RECEPTOR>_mig_cdr3_mapping_stats.csv`: mapping statistics for all samples of the receptor type

- **Subfolders**

Each subfolder is named for and corresponds to the `sampleID` value for the samples defined in the metadata input file. The subfolder of each sample contains multiple CSV files, corresponding to receptor chains.

- For BCRv1, there are four files: for IGG, IGK, IGL, and IGM
- For BCRv2, there are seven files: for IGA, IGD, IGE, IGG, IGK, IGL, and IGM
- For TCRv2, there are two files: for TRA and TRB

These files are identical to the clonotype worksheets (individual tabs) included in the `<RECEPTOR>_<sampleID>_mig_cdr3_report.xlsx` file. They are provided in CSV format to make it easier for advanced users to import them into other tools for tertiary analysis.

Appendix B. Log Files

Table 13. Immune Profiler log files. The table contains the filename, the location where to find them within the `immune_profiler/` directory, and a brief description of the information stored in each.

Log filename	Subfolder	Description
<code><output name>_immune_profiler.log</code>	--	Immune Profiler analysis progress and important notes
<code>mig_run_migec.log</code>	<code>run_migec/</code>	MIGEC analysis progress
<code>mig_run_migec.error</code>	<code>run_migec/</code>	MIGEC error messages (if any)
<code>assemble.log.txt</code>	<code>run_migec/assemble</code>	MIGEC process status
<code>assemble.cmd.txt</code>	<code>run_migec/assemble</code>	The command call to MIGEC for the assemble function
<code>checkout.cmd.txt</code>	<code>run_migec/checkout_all/</code>	MIGEC analysis commands used
<code>checkout.log.txt</code>	<code>run_migec/checkout_all/</code>	MIGEC analysis progress and findings
<code>checkout.filelist.txt</code>	<code>run_migec/checkout_all/</code>	Documents all intermediate files while MIGEC is processing
<code>run_mixcr_[cdr3 fl].log</code>	<code>run_mixcr/</code>	MIXCR analysis progress and important notes. NOTE: Different files are created depending on whether the CDR3 or Full_length target regions are selected during configuration
<code>run_mixcr_[cdr3 fl].error</code>	<code>run_mixcr/</code>	MIXCR error messages (if any) NOTE: Different files are created depending on whether the CDR3 or Full_length target regions are selected during configuration

Appendix C. More Output Details from Immune Profiler

A. Preprocess Folder

This folder is generated for results of the first step of the Immune Profiler workflow ([Section III](#)), which separates reads in original sample-level FASTQs into different chain-specific FASTQs.

- FASTQs are created for each of the chain types.
 - BCRv1 <chain type> : IgG, IgM, IgK, and IgL
 - BCRv2 <chain type> : IgA, IgD, IgE, IgG, IgM, IgK, and IgL
 - TCRv2 <chain type> : TRA and TRB

`<sampleID>_<chain type>_R1.fastq` and `<sampleID>_<chain type>_R2.fastq`
- An undetermined FASTQ pair is generated to store reads that cannot be confidently assigned to any chain categories.

`<sampleID>_undetermined_R1.fastq`
`<sampleID>_undetermined_R2.fastq`
- Reads that are less than 30 bases in length, too short to be accurately aligned with any V(D)J sequences, are assigned to:

`<sampleID>_short_R1.fastq`
`<sampleID>_short_R2.fastq`
- If linker-based correction is turned on, an additional FASTQ pair is created to store reads that failed to correct:

`<sampleID>_flc_R1.fastq`
`<sampleID>_flc_R2.fastq`

REMINDER: If the "Keep intermediate file?" option is not selected, the `preprocess/` folder is deleted by Immune Profiler to save storage space and computing resources on the workstation or server.

B. run_migec Folder

This folder is created during the second step of the Immune Profiler workflow. A version of MIGEC embedded in Immune Profiler is deployed to conduct error correction using Unique Molecular Identifiers (UMIs). Algorithm details for this processing can be found at <https://migec.readthedocs.io/en/latest/index.html>. If you have additional questions after referring to the documentation, please contact technical_support@takarabio.com.

- `barcodes.txt` : an intermediate file generated by Immune Profiler to link sample information with MIGEC processing
- `assemble/` : reads from the same UMI are grouped together and defined as a Molecular Identifier Group (MIG). Reads within the same MIG are cross-referenced, potential sequence errors are corrected, and collapsed reads are deducted. The `assemble/` folder stores the resulting FASTQs.

`<sampleID>_<chain type>_R1.t*.cf.fastq`
`<sampleID>_<chain type>_R2.t*.cf.fastq`

For t^* in the FASTQ file name, the $*$ represents a numeric value; the value indicates the UMI threshold used for the corresponding chain-specific samples.

IMPORTANT: Immune Profiler assumes UMIs are on Read 1, while the SMARTer Human BCR IgG IgM H-K-L Profiling Kit v1, SMARTer Human BCR IgA IgD IgE IgG IgM H-K-L Profiling Kit v2, SMART-Seq Human TCR (with UMIs), and SMARTer Human TCR a/b Profiling Kit v2 chemistry have UMIs on Read 2. The FASTQ file names, therefore, correspond to their opposite read. Example:

- The file `<sampleID>_<chain type>_R1.T1.CF.FASTQ` is FASTQ Read 2
- `<sampleID>_<chain type>_R2.t1.cf.fastq` is Read 1

- `checkout_all/` : read screening is performed and trustable reads deducted, creating sample-level clean-up FASTQs.

`<sampleID>_<chain type>_R1.fastq` and `<sampleID>_<chain type>_R2.fastq`

Any undetermined reads identified based on UMI algorithms are assigned to these files:

`undef-R1.fastq` and `undef-R2.fastq`

NOTE: If the "Keep intermediate file" option is not selected, all the FASTQs in this folder are deleted after processing.

- `histogram/` : UMI statistics are collected, and a threshold is determined for read exclusion (Shugay et al. 2014). In total 10 files are created at this step; details about these files can be found at <https://migec.readthedocs.io/en/latest/logs.html>. If you have additional questions after referring to the documentation, please contact technical_support@takarabio.com.
 - o `estimates.txt`
 - o `histogram.cmd.txt`
 - o `overseq.txt`
 - o `overseq-units.txt`
 - o `collision1.txt`
 - o `collision1-units.txt`
 - o `pwm.txt`
 - o `pwm-units.txt`
 - o `pwm-summary.txt`
 - o `pwm-summary-units.txt`

C. run_mixcr Folder

This folder results from the third step of the analysis workflow. A version of MiXCR embedded in Immune Profiler is called to conduct read alignment, assembling, and clonotype reporting. Algorithm details can be found at https://mixcr.readthedocs.io/_/downloads/en/docs/pdf/ (PDF). If you have additional questions after referring to the documentation, please contact technical_support@takarabio.com.

Depending on the target region specified in the optional configuration arguments, the folders `mig_cdr3/`, `mig_fl/`, or both folders may be created to store corresponding analysis results.

Each folder will contain files with the following:

- Sample- and chain-level alignment statistics report
`<sampleID>_<chain type>_mig_[cdr3|fl]_align_report.txt`
- Sample- and chain-level clonotype assembling statistics report
`<sampleID>_<chain type>_mig_[cdr3|fl]_clones_report.txt`
- Sample- and chain-level clonotype detail report
`<sampleID>_<chain type>_mig_[cdr3|fl]_clones_all.txt`
- Two files in binary format are created for each sample and chain.
`<sampleID>_<chain type>_mig_[cdr3|fl].vdjca`
`<sampleID>_<chain type>_mig_[cdr3|fl].clns`

The string `[cdr3|fl]` means either `cdr3` or `fl` will be inserted there, as in the following example:

`<sampleID>_<chain type>_mig_cdr.vdjca`
`<sampleID>_<chain type>_mig_fl.clns`

NOTE: If the option to keep intermediate files is selected, the *.vdjca and *.clns files are retained. By default, they are deleted after processing.

Contact Us	
Customer Service/Ordering	Technical Support
tel: 800.662.2566 (toll-free)	tel: 800.662.2566 (toll-free)
fax: 800.424.1350 (toll-free)	fax: 800.424.1350 (toll-free)
web: takarabio.com/service	web: takarabio.com/support
e-mail: ordersUS@takarabio.com	e-mail: technical_support@takarabio.com

Notice to Purchaser

Our products are to be used for **Research Use Only**. They may not be used for any other purpose, including, but not limited to, use in humans, therapeutic or diagnostic use, or commercial use of any kind. Our products may not be transferred to third parties, resold, modified for resale, or used to manufacture commercial products or to provide a service to third parties without our prior written approval.

Your use of this product is also subject to compliance with any applicable licensing requirements described on the product's web page at takarabio.com. It is your responsibility to review, understand and adhere to any restrictions imposed by such statements

© 2022 Takara Bio Inc. All Rights Reserved.

All trademarks are the property of Takara Bio Inc. or its affiliate(s) in the U.S. and/or other countries or their respective owners. Certain trademarks may not be registered in all jurisdictions. Additional product, intellectual property, and restricted use information is available at takarabio.com.

This document has been reviewed and approved by the Quality Department.