Takara Bio USA, Inc.

Cogent™ NGS Immune Profiler User Manual

Cat. Nos. Many
software v2.0
(111524)

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

Table of Contents

Table of Figures

Table of Tables

I. Introduction

Cogent NGS Immune Profiler Software (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 and mouse 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). Immune Profiler incorporates a third-party program, TRUST4, packaged and included for use 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.

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.

A. What's New

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

- **Cogent NGS Immune Profiler v2.0**
  - Incorporates the TRUST4 immune profiling algorithm
  - Newly supported kits: SMART-Seq® Mouse BCR (with UMIs), SMARTer® Human TCR a/b Profiling Kit, and SMARTer Mouse TCR a/b Profiling Kit
  - Improved algorithms for linker correction and isotype-specific primer detection
  - Improved handling of large FASTQ files
  - Customizable usage control of server CPU resources while running the pipeline
  - Updated download procedure—reversion to the procedure for CogentIP v1.6 and prior

NOTE: Find release notes for prior versions on the Cogent NGS Immune Profiler product page.

B. Supported Immune Profiling Products

Table 1 lists all the Takara Bio immune profiling products that the sequencing results of which can be processed by CogentIP.

Table 1. Immune profiling products supported by Cogent NGS Immune Profiler.

Species	Type	Product name	Catalog No.
Human	BCR	SMART-Seq Human BCR (with UMIs)	634776, 634777 & 634778
		SMARTer Human BCR IgG IgM H/K/L Profiling Kit	634466 & 634467
	TCR	SMART-Seq Human TCR (with UMIs)	634779, 634780 & 634781
		SMARTer Human TCR a/b Profiling Kit v2	634478 & 634479
		SMARTer Human TCR a/b Profiling Kit	635016
Mouse	BCR	SMART-Seq Mouse BCR (with UMIs)	634351, 634352 & 634353
	TCR	SMART-Seq Mouse TCR (with UMIs)	634814, 634815 & 634816
		SMARTer Mouse TCR a/b Profiling Kit	634402*, 634403 & 634404

*Catalog number is discontinued.

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 2.5 x 10^7 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. User Account Requirements

The account used to install Immune Profiler needs to have read/write (R/W) permissions for the following folders:

- Where the Immune Profiler program will be located,
- Where Profiler will be run, and
- Where the analysis outputs will be saved.

Once installed, other user accounts can be used to run the Immune Profiler executable, but these accounts need to have R/W permissions for the latter two folders, above.

D. Additional Hardware and Software Dependencies and Recommendations

- **Java 11 or higher**

  NOTE: This will NOT work with Java SE 10 or lower. If running a version lower than Java 11 on the target install server, please upgrade. Uninstall the earlier Java version and install a supported version described above by downloading the installation executable from Oracle.

- **Conda 23.7.4 or higher**

  If Conda is not currently installed on the server, instructions to do so can be found at https://conda-forge.org/download/.

- **Internet connectivity on the server**

  Under certain circumstances, internet connectivity is required when running Immune Profiler:

  - The first time it is run from a given directory (Section V.A, "Best Practices")
  - If a different version of TRUST4 is specified by the run options (Section V.B.3, "Performance Configuration Arguments")
  - If the `work/conda/` directory is deleted. See Section VI.A, "Output Overview", for more information on this folder.

  If none of the conditions above apply to your current run, then CogentIP can be run offline/without a connection to the Internet.

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 1 x 10^7 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`. Immune Profiler will check and make sure these specified files exist; if no file matching the name is found, an error message is displayed, prompting the user to double-check the FASTQ file names.

An example of the metadata file contents is shown in Table 2 (next page).

Table 2. 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 the mini dataset samples included with the software. See Appendix A for more information about the mini dataset samples and result files.

III. Software Overview

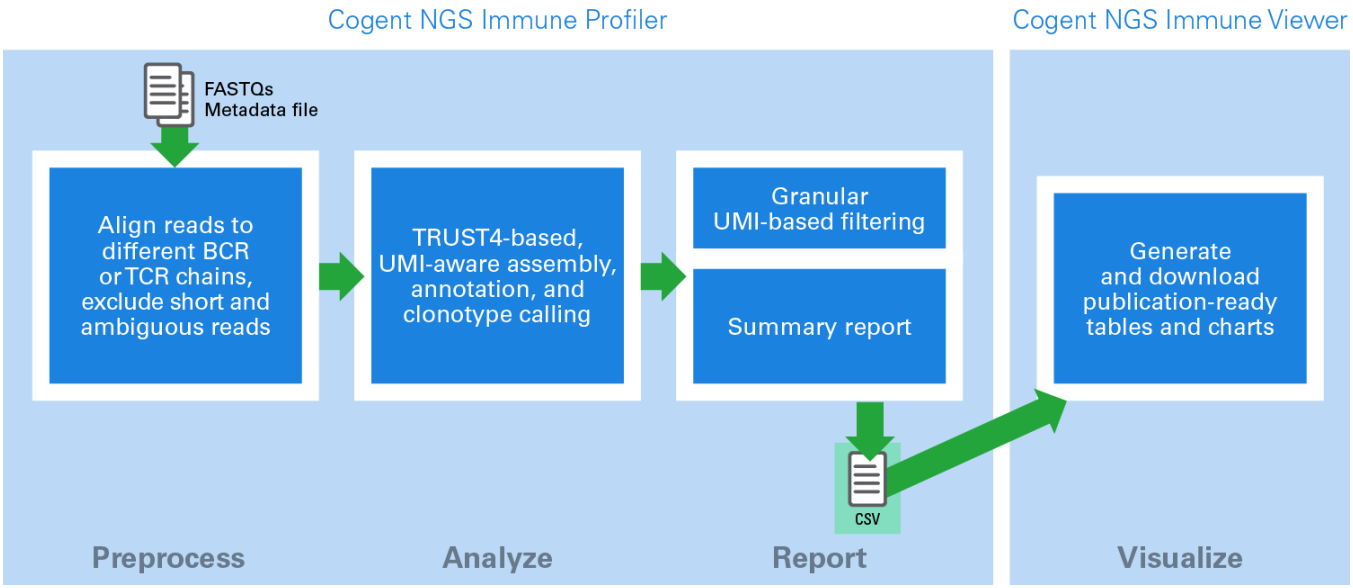

Figure 1. Cogent NGS Immune Profiler analysis workflow.

The CogentIP workflow consists of four steps and 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
   - Split preprocessed FASTQ files into multiple chunks to improve processing time. UMI groups are kept intact rather than being split across multiple FASTQ files.
2. Analysis (TRUST4 based clonotype analysis):
   - Aligns reads from FASTQ file to the VDJ sequences and performs de novo assembly to construct full length immune receptor sequences. This is followed by contig assembly and clonotype identification.
   - Merge TRUST4 analysis chunks that are used to generate final reports.
3. Summary reports:
   - Summarizing read QC statistics: chain-specific, short, undetermined, failed linker-based correction (flc)

- Summarizing clonotype details: numbers, percentage, nucleotide, and amino acid sequence (compatible with Cogent NGS Immune Viewer)
- Custom UMI cutoffs can be used to generate reports with clonotypes identified using UMIs from UMI groups of a minimum size.

IV. Installation and Configuration Requirements

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

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 v2.0.

Follow the uninstall directions in Section IV.G ("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

CogentIP 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_v2.0.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 necessary, move or copy the CogentIP ZIP file onto the Linux server or Mac into the directory location where you want to install CogentIP.

   NOTE: The account logged into while doing the installation must have read/write privileges to the install directory chosen.

3. From the same directory location in Step 2, unzip the CogentIP ZIP file. On Linux, this can be done by running the following two commands in the order listed:

   ```
   unzip Cogent_NGS_Immune_Profiler_v2.0.zip
   ```

   ```
   cd immune_profiler
   ```

   The following files should be included in the `immune_profiler/` directory (Figure 2):

   - `Cogent NGS Immune Profiler User Manual.pdf.`
   - `Cogent NGS Immune Profiler and Cogent NGS Immune Viewer Quick Start Guide.pdf.`
   - `VERSION`: file with software version information
   - `bin/`: folder storing main analysis scripts.
   - `main.nf`, `main_umi_reports.nf`, `modules`, `nextflow.config`, `cogentip.yaml`, and `share`: files related to the Nextflow program required by CogentIP
   - `test/` : folder contains a directory of test dataset files (`test_input/`) and a directory of example outputs generated by the test dataset files (`test_output/`). More information about this folder can be found in Appendix A.

```
immune_profiler/
├── bin/
├── modules/
├── share/
├── test/
├── Cogent_NGS_Immune_Profiler_User_Manual.pdf
├── Cogent_NGS_Immune_Profiler_and_Cogent_NGS_Immune_Viewer_Quick_Start_Guide.pdf
├── README.md
├── VERSION
├── cogentip.yaml
├── main.nf
├── main_umi_reports.nf
└── nextflow.config
```

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

C. (Optional) Set Up $PROFILER_HOME Environmental Variable

For ease of use, we recommend that the CogentIP install directory location be added to the `.bash_profile` as a permanent environmental variable.

Example:

If your account name is 'myacct', the absolute pathname for myacct's home directory is `/home/myacct`, which has a UNIX alias of `~/`. If CogentIP was installed following the instructions in Section IV.B into the `~/bin/` directory, edit `.bash_profile` to add the following line:

```
export PROFILER_HOME=/home/myacct/bin/immune_profiler
```

Once added to the profile, you will either need to log out and back into the account, or load the file in with the command:

```
source ~/.bash_profile
```

The phrase `$PROFILER_HOME` can then be used in place of :

```
/home/myacct/bin/immune_profiler/
```

Example:

Running the following while logged in as 'myacct' will change directory to `~/bin/immune_profiler/`:

```
cd $PROFILER_HOME
```

NOTE: Subsequent references to `$PROFILER_HOME` in this document refer to the full path where Immune Profiler is installed.

D. 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 "20" 2023-03-21
OpenJDK Runtime Environment Homebrew (build 20)
OpenJDK 64-Bit Server VM Homebrew (build 20, mixed mode, sharing)
```

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 version is 11 or higher, up to 23.

For assistance installing Java, visit the website:

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

E. Verifying conda Installation and Version

To verify that Conda is installed properly on the server.

1. Type the following command in at the prompt in any directory location on the server.

   ```
   conda -V
   ```

   If Conda is successfully installed, it should return text with the version number.

   e.g.,

   ```
   conda 23.7.4
   ```

2. Check to see if the base Conda environment can be activated. Type the following command into the command-line prompt on the Linux server:

   ```
   conda activate
   ```

   A successful Conda install will result in a change in the prompt, as shown in Figure 4.

```
$ conda activate
(base) $
(base) $ conda deactivate
$
```

Figure 4. Screenshot of the Linux command line showing a successful check of the base Conda environment. If the Conda installation was not completed as required, both commands would return error messages.

3. If Conda is successfully installed and the prompt changed as displayed in Figure 2, type the following command to return to the default Linux prompt:

   ```
   conda deactivate
   ```

   This command will take you back to the Linux prompt and out of the Conda environment.

4. Installation of miniforge3 typically adds the location of its installation to the user's system environment. This is also required for the successful installation of CogentIP.

   The following steps can be used to verify that the Conda `$PATH` is configured correctly.

   a. Open the file `.bash_profile`, which for an individual user account will be located in the home directory:

      ```
      more ~/.bash_profile
      ```

   b. Verify a line similar to the following is showing in the file:

      ```
      export PATH="/home/<USERNAME>/miniforge3/bin:$PATH"
      ```

      where `<USERNAME>` is replaced by the username of the account that installed Conda.

      e.g., username is 'myacct':

      ```
      export PATH="/home/myacct/miniforge3/bin:$PATH"
      ```

If the line isn't displaying or the `.bash_profile` file does not exist, it will need to be manually created and populated. For more information on setting an environment variable, see a UNIX user manual or a forum post like https://stackoverflow.com/a/7502128.

F. Conduct Test Run with the Mini Datasets

After installation, an analysis should be done to install the conda-based software dependencies and to verify the install using one of the mini dataset sample files provided in `test/test_input/`. See Appendix A for more information about the mini dataset and results files.

Table 3 can be used as an aid to determine which mini dataset may be of most interest to you.

Table 3. Immune profiling kit prefix reference, mini dataset data directory, and mini dataset metadata file names.

Reagent kit	Prefix	Mini dataset data directory	Mini dataset results directory
SMART-Seq Human BCR (with UMIs)	hBCRv2	hBCRv2_mini	hBCRv2_mini_results
SMART-Seq Human TCR (with UMIs)	hTCRv2	hBCRv2_mini	hTCRv2_mini_results
SMART-Seq Mouse BCR (with UMIs)	mBCRv2	mTCRv2_mini	mBCRv2_mini_results
SMART-Seq Mouse TCR (with UMIs)	mTCRv2	mTCRv2_mini	mTCRv2_mini_results

NOTE: The six-character prefix value (abbreviation) in Table 3 will be used throughout the manual as shorthand for the output of the listed reagent kits, in terms of options within the software.

1. 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.
2. Follow the steps in Section V.B.4, "Command Line Examples", to set up and execute an analysis run on your system.
3. Compare the analysis results (1) generated by your run to the results provided in the `reports/` subfolder (Section VI.C) to (2) the mini dataset output in `test/test_output/`**`<species>`**`/`**`<prefix>`**`_mini_results`.

The installation is considered to be successful if the output to your test run matches the results stored in `test/test_output/`**`<species>`**`/`**`<prefix>`**`_mini_results/` (e.g., `test/test_output/human/hBCRv2_mini_results/`).

G. Uninstalling Immune Profiler

1. Move any output files that you want to keep that are 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 to upgrade to a newer version, return to Section IV.B, "Immune Profiler Download and Installation", to continue.

V. Using Immune Profiler

A. Best Practices

- After the initial installation or the first time running CogentIP from a given directory, run the Immune Profiler help command to install the Immune Profiler dependencies.

```
$PROFILER_HOME/bin/cogentip analyze -h
```

- 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.F, "Conduct Test Run with the Mini Datasets").
- The computer Cogent NGS 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, "Processing Time"), this recommendation is to prevent the computer from shutting down before the analysis is finished.

B. Run Immune Profiler

Immune Profiler is launched via command line interface (CLI) utilizing the **`cogentip`** script. This script can be launched either from within the `immune_profiler/bin/` directory or from any location (working directory) on the computer where Immune Profiler software is installed if the full path to the script is specified or using an environmental variable (Section IV.C).

```
$PROFILER_HOME/bin/cogentip analyze -r <RECEPTOR> -s <SPECIES> -m
<METADATA> -o <OUTPUT>
```

The full list of arguments can be accessed with the `-h` option (Figure 5):

```
$PROFILER_HOME/bin/cogentip analyze -h
```

A fuller explanation of the variables is provided after Figure 5.

```
$ bin/cogentip analyze -h
usage: cogentip analyze [-h] [-V] -r {TCRv1,TCRv2,BCRv1,BCRv2} -s {human,mouse} -m META_FILE -o OUTDIR
                        [-l] [-u UMI_CUTOFF] [--ignore_umis] [--trust4 TRUST4_VERSION] [--threads THREADS]
                        [-c CHUNKS] [-k {0,1,2,3,4}] [--resume]

informational arguments:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit

required arguments:
  -r {TCRv1,TCRv2,BCRv1,BCRv2}, --receptor_kit {TCRv1,TCRv2,BCRv1,BCRv2}
                        the receptor kit
  -s {human,mouse}, --species {human,mouse}
                        the species of the sample
  -m META_FILE, --meta_file META_FILE
                        path to the file containing sample IDs and corresponding FASTQ pairs
  -o OUTDIR, --outdir OUTDIR
                        path to the output directory

optional arguments:
  -l, --linker_correction
                        remove reads that do not have the linker sequence in the expected location (default: False)
  -u UMI_CUTOFF, --umi_cutoff UMI_CUTOFF
                        UMI group size cutoff (default: 1)
  --ignore_umis         ignore UMIs during analysis of UMI-based data (default: False)

performance arguments:
  --trust4 TRUST4_VERSION
                        the TRUST4 version to use in X.Y.Z format (minimum: 1.1.4) (default: 1.1.4)
  --threads THREADS     the number of threads to use for FASTQ Extraction and Contig Annotation (default: 4)
  -c CHUNKS, --chunks CHUNKS
                        the number of chunks to split FASTQ files into for assembly (default: 4)
  -k {0,1,2,3,4}, --keep_intermediate {0,1,2,3,4}
                        keep some/all intermediate files (Increases run time and storage requirements) (default: 0)
  --resume              resume analysis after error (default: False)
```

Figure 5. Output for the `cogentip -h` command.

There are four types of arguments:

1. Required information (Section V.B.1)
2. Optional configuration (Section V.B.2)
3. Performance configuration (Section V.B.3)
4. Informational—the arguments `-h` for help and `-v` to display the Immune Profiler version.

Users must specify all required arguments to launch Immune Profiler. The optional and performance arguments have default values and can be omitted or included based on the analysis needs.

1. Required Information Arguments

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

- `-r, --receptor_kit` : the receptor type of the data files to be analyzed: `BCRv1`, `BCRv2`, `TCRv1,TCRv2`

  Example:

  If the input files represent data for samples from SMART-Seq Human TCR (with UMIs),

SMART-Seq Mouse TCR (with UMIs), or SMARTer Human TCR a/b Profiling Kit v2, then this argument and parameter would be typed:

```
-r TCRv2
```

- `-s, --species` : the species of the sample (options: human or mouse)
- `-m, --meta_file` : the path to and name of the metadata file (described in Section II.E)

  NOTE: FASTQ files configured in the metadata file should be represented as absolute paths or as paths relative to the location of the metadata file. CogentIP is case-sensitive.

- `-o, --outdir` : output directory

  This is the path to the output directory. The directory will be created by the software and should not already exist.

2. Optional Configuration Arguments

- `-l, --linker_correction` : specify whether 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, --umi_cutoff` : specify an integer to use as the UMI cutoff.

  (default: 1)

  NOTE: This parameter sets the same UMI cutoff for all chain types of each sample.

  The optimal value is dependent on sequencing depth and chain type abundance within the sample. To increase the confidence of the analysis result, the value can be increased; however, values that are too high may result in insufficient reads per UMI to obtain meaningful data.

  Alternatively, custom sample-by-chain type-specific UMI cutoffs can be used to generate a new set of reports after the initial analysis. To aid in choosing appropriate UMI cutoffs, a set of figures (`umi_stats/umi_group_sizes_frequency.<SAMPLE_NAME>.png`, where **`<SAMPLE_NAME>`** is the name you've given to the sample) is generated for each sample that shows the UMI and UMI group frequencies observed per UMI group size for each chain type. Based on these figures, the UMI cutoffs template (`umi_stats/umi_cutoffs.template.csv`) can be populated and used with the `cogentip report` command described in Section V.C (below).

- `--ignore_umis` : ignore UMIs during the analysis of UMI-based data (default: false)

3. Performance Configuration Arguments

- `--trust4` : minimum TRUST4 version to use to run the analysis in `#.#.#` format
  (default: 1.1.4)
  1.1.4 is also the minimum version that should be used.
- `--threads` : specify number of threads to use for TRUST4 steps that can run on multiple threads
  (default: 4)
- `-c, --chunks` : specify number of analysis chunks to split FASTQ files into
  (default: 4)
  This option helps to reduce runtime of analysis
- `-k, --keep_intermediate` : level of intermediate files to keep
  (default: 0)

  During the analysis, Immune Profiler splits the input preprocessed FASTQ files into chunks to speed up processing. Some of the results from these chunks are merged for further downstream processing; other results from these chunks do not need to be merged for the final files. This parameter provides the option to merge such results files.

  Retaining some or all of the intermediate files is optional since merging takes additional time and disk space, as these files are large. The value specified with this option controls the quantity (or level) of the intermediate files which are retained.

  The five options for this variable are 0, 1, 2, 3, or 4. Each option is incremental and includes the files retained from the levels below it.

  - `0` : All intermediate files are deleted once analysis has been completed.
  - `-k` ≥ 1 : The preprocessed FASTQ files are retained, the rest are deleted.
  - `-k` ≥ 2 : The TRUST4 results from each analysis chunk are retained
  - `-k` ≥ 3 : The FASTQ files from each analysis chunk are retained (in `split_fastq_files`)
  - `4` : The intermediate TRUST4 results from each analysis chunk that are not required to generate final reports are merged and retained

  Example:

  If `-k 3` is specified, the following files are retained:

  - preprocessed FASTQ files (`-k`≥1)
  - the TRUST4 results from each analysis chunk (`-k`≥2)
  - the FASTQ files from each analysis chunk (`-k`≥3)

  However, the intermediate TRUST4 results from each analysis chunk that are not required to generate final reports are *not* retained, as '3' is not greater than or equal to '4'.

- `--resume` : Resumes analysis run that has been stopped or that encountered an error.

  When using `--resume`, please keep all other parameters the same as the previous analysis run that is being resumed. The one exception is that the number of threads can be adjusted by using the `--threads` parameter. Resuming an analysis run will not resolve

all errors but can help in instances where errors result from issues with computational resources (e.g., running out of memory or storage space).

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 4 or Table 5:

Table 4. Human BCRv2 and TCRv2 mini dataset samples analysis parameters to match the associated sample output.

Parameters	BCRv2	TCRv2
Receptor Kit (`-r`)	`BCRv2`	`TCRv2`
Species (`-s`)	`human`	`human`
Metadata File (`-m`)	`test/test_input/human/hBCRv2_mini/hBCRv2_mini_meta.csv`	`test/test_input/human/hTCRv2_mini/hTCRv2_mini_meta.csv`
Output Directory (`-o`)	`hBCRv2_mini_results`	`hTCRv2_mini_results`

Table 5. Mouse BCRv2 and TCRv2 mini dataset samples analysis parameters to match the associated sample output.

Parameters	BCRv2	TCRv2
Receptor Kit (`-r`)	`BCRv2`	`TCRv2`
Species (`-s`)	`mouse`	`mouse`
Metadata File (`-m`)	`test/test_input/mouse/mBCRv2_mini/mBCRv2_mini_meta.csv`	`test/test_input/mouse/mTCRv2_mini/mTCRv2_mini_meta.csv`
Output Directory (`-o`)	`mBCRv2_mini_results`	`mTCRv2_mini_results`

NOTE: For the examples in a)–d) below, the command text should be typed or pasted onto one CLI prompt.

a) Human BCRv2

To analyze the human BCRv2 mini dataset and generate identical report files to `$PROFILER_HOME/test/test_output/human/BCRv2_mini_results`.

```
$ $PROFILER_HOME/bin/cogentip analyze \
  -r BCRv2 \
  -s human \
  -m $PROFILER_HOME/test/test_input/human/hBCRv2_mini/hBCRv2_mini_meta.csv \
  -o hBCRv2_mini_results
```

b) Human TCRv2

To analyze the human TCRv2 sample dataset and generate identical report files to `$PROFILER_HOME/test/test_output/human/TCRv2_mini_results`

```
$ $PROFILER_HOME/bin/cogentip analyze \
  -r TCRv2 \
  -s human \
  -m $PROFILER_HOME/test/test_input/human/hTCRv2_mini/hTCRv2_mini_meta.csv \
  -o hTCRv2_mini_results
```

c) Mouse BCRv2

To analyze the mouse BCRv2 sample dataset and generate identical report files to `$PROFILER_HOME/test/test_output/mouse/mBCRv2_mini_results`

```
$ $PROFILER_HOME/bin/cogentip analyze \
  -r BCRv2 \
  -s mouse \
  -m $PROFILER_HOME/test/test_input/mouse/mBCRv2_mini/mBCRv2_mini_meta.csv \
  -o mBCRv2_mini_results
```

d) Mouse TCRv2

To analyze the mouse TCRv2 sample dataset to generate identical report files to `$PROFILER_HOME/test/test_output/mTCRv2_mini_results`

```
$ $PROFILER_HOME/bin/cogentip analyze \
  -r TCRv2 \
  -s mouse \
  -m $PROFILER_HOME/test/test_input/mouse/mTCRv2_mini/mTCRv2_mini_meta.csv \
  -o mTCRv2_mini_results
```

C. Generating Reports with Custom UMI Cutoffs

As mentioned in Section V.B.2, "Optional Configuration Arguments", by default, final reports are generated using a UMI cutoff `(-u)` value of '1' (i.e., no filtering based on UMI group size).

However, for users wanting to fine-tune their data, CogentIP has a more advanced command, `cogentip report`, to assist them with doing that.

```
$PROFILER_HOME/bin/cogentip report \
-a <PREVIOUS_ANALYSIS_OUTPUT_DIRECTORY> \
-u <FULL_PATH_UMI_CUTOFFS_CSV> \
-s <SPECIES>
```

The full list of arguments can be accessed with the `-h` option (Figure 6):

```
$PROFILER_HOME/bin/cogentip report -h
```

```
$ bin/cogentip report -h
usage: cogentip report [-h] [-V] -a ANALYSIS_DIR -u UMI_CUTOFFS -s {human,mouse} [--trust4 TRUST4_VERSION] [--resume]

informational arguments:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit

required arguments:
  -a ANALYSIS_DIR, --analysis_dir ANALYSIS_DIR
                        path to the top-level directory of a previous CogentIPv2 analysis
  -u UMI_CUTOFFS, --umi_cutoffs UMI_CUTOFFS
                        path to the UMI cutoffs CSV file
  -s {human,mouse}, --species {human,mouse}
                        the species of the sample

performance arguments:
  --trust4 TRUST4_VERSION
                        the TRUST4 version to use in X.Y.Z format (minimum: 1.1.4)
  --resume              resume analysis after error
```

Figure 6. Output for the `cogentip report -h` command.

There are two types of input data:

1. The output folder of a previous analysis run (`cogent analyze`)—this is the same folder specified by the `-o (--outdir)` required argument.
2. A `umi_cutoffs.template.csv` file (see Section V.C.1.b)—this initial file is autogenerated during the `cogent analyze` run

and three types of arguments:

- Required information (see Section V.C.2)—all required arguments must be specified to use this command
- Performance configurations (see Section V.C.3)
- Informational—the arguments -h for help and -v to display the `cogentip report` version.

The sections below contain background information on how to approach using this command for your experiment.

1. Supporting Files

a) UMI and UMI Group Frequencies Plots (umi_group_sizes_frequency)

To assist users in customizing the UMI cutoff value to fit their data, a set of PNG file plots are generated in the `umi_stats/` folder with the naming convention of

`umi_group_sizes_frequency.<sampleID>.png`

where `<sampleID>` is the value of the corresponding sample ID as defined in the metadata file (Section II.E.2). The plot shows the UMI and UMI group frequencies observed per UMI group size for each chain type.

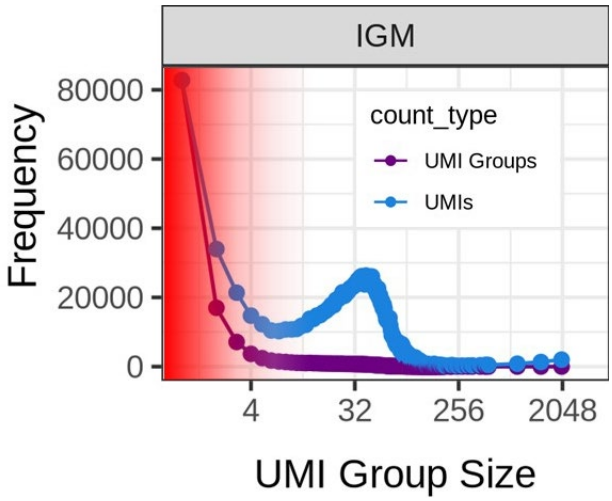

Figure 7. Example `umi_group_sizes_frequency.<sampleID>.png` plot.

In the example shown in Figure 7, lower confidence UMI groups (shaded in red) are more likely to be the result of sequencing errors. To increase the confidence of the analysis result, the UMI cutoff value can be increased. However, a UMI cutoff value that is too high may result in insufficient reads per UMI to obtain meaningful data. An optimal UMI cutoff will remove small UMI Groups that have inflated frequencies relative to the moderately size UMI Groups.

b) Custom UMI Cutoff Values by Sample and Chain Type

Based on these plots, the autogenerated UMI cutoffs template (`umi_stats/umi_cutoffs.template.csv`) can be modified and used with the `cogentip report` command. Figure 8 shows an example of what the CSV file might look like when opened in Microsoft Excel.

	A	B	C	D
1	sample	isotype	umi_cutoff	
2	S4	IGA		
3	S4	IGE		
4	S5	IGA		
5	S5	IGD		
6	S5	IGL		
7	S6	IGA		
8	S6	IGE		
9	S6	IGG		
10	S6	IGL		
11	S6	IGM		
12				
13				

Figure 8. Example `umi_cutoffs.template.csv` file contents.

The file contains three columns of information:

[A] sample—(default: prepopulated) this refers to the "sampleID" assigned previously to the experimental data and originally referenced in the metadata file (Section II.E.2)

[B] isotype—(default: prepopulated) the molecular structure of a chain type within the sample pool (e.g., IgA, IgG, IgK, etc.)

[C] umi_cutoff—(default: blank) the numerical value that should be used by `cogentip report` for the UMI cutoffs value, equivalent to the `-u, --umi_cutoff` argument in the analyze command (Section V.B.2, "Optional Configuration Arguments")

To use the `cogentip report` function, you will need to edit this file to fill in the umi_cutoff value (Column C) for each row (sampleID+isotype combination) of interest in the file. If the umi_cutoff column is not populated for a given row, then cogentip report will not include that combination in the new report.

As a reminder:

- The default value is '1'
- The value you use could be '1' if you do not want to change the analysis parameters of that particular sampleID+isotype combination but want to include it in the new report.
- To increase the confidence of the analysis result, the UMI cutoff value should be increased. However, a UMI cutoff value that is too high may result in insufficient reads per UMI to obtain meaningful data.
- An optimal UMI cutoff will remove small UMI Groups inflated frequencies relative to the moderately size UMI Groups.

2. Required Information Arguments

See the example command below for how these arguments might be configured.

- `-a, --analysis_dir` : the top-level output directory of a previous Immune Profiler analysis. In the original profiler run (Section V.B), this is the value specified by the `-o` (`--outdir`) argument.
- `-u, --umi_cutoffs` : the path to the UMI cutoffs CSV file
- `-s, --species` : the species of the sample (options: human or mouse)

3. Performance Configuration Arguments

`--resume` : Resumes analysis run that has been stopped or that encountered an error.

When using `--resume`, please keep all other parameters the same as the analysis run that is being resumed. Resuming an analysis run will not resolve all errors but can help in instances where errors result from issues with computational resources (e.g., an error resulting from running out of memory or storage space).

4. Command Line Example

To run the `cogentip report` command using the sample output included with CogentIP, choose the parameters for the desired receptor type from the appropriate column in Table 6 or Table 7:

Table 6. Human BCRv2 and TCRv2 mini dataset sample parameters for the `cogentip report` command.

Parameters	BCRv2	TCRv2
Previous analysis output directory (`-a`)	`test/test_output/human/mBCRv2_mini/`	`test/test_output/human/mTCRv2_mini/`
Full path and file name to the `umi_cutoffs.template.csv` file (`-u`)	`test/test_output/human/mBCRv2_mini/umi_stats/umi_cutoffs.template.csv`	`test/test_output/human/mTCRv2_mini/umi_stats/umi_cutoffs.template.csv`
Species (`-s`)	`human`	`human`

Table 7. Mouse BCRv2 and TCRv2 mini dataset sample parameters for the `cogentip report` command.

Parameters	BCRv2	TCRv2
Previous analysis output directory (`-a`)	`test/test_output/mouse/mBCRv2_mini/`	`test/test_output/mouse/mTCRv2_mini/`
Full path and file name to the `umi_cutoffs.template.csv` file (`-u`)	`test/test_output/mouse/mBCRv2_mini/umi_stats/umi_cutoffs.template.csv`	`test/test_output/mouse/mTCRv2_mini/umi_stats/umi_cutoffs.template.csv`
Species (`-s`)	`mouse`	`mouse`

For the example below, with the required arguments based on the mTCRv2 Mouse sample output, the command text should be typed or pasted onto one CLI prompt.

```
$ $PROFILER_HOME/bin/cogentip report \
  -a $PROFILER_HOME/test/test_output/mouse/mTCRv2_mini/ \
  -u $PROFILER_HOME/test/test_output/mouse/mTCRv2_mini/umi_stats/umi_cutoffs.template.csv \
  -s mouse
```

D. Processing Time

The runtime 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 8. Dataset parameters used for benchmark testing.

Dataset parameters	Dataset 1 (small)	Dataset 2 (large)
Number of samples	4	12
Read pairs per sample	1 million PE301	20 million PE301
Aggregate FASTQ file size	782 MB	45 GB

The following two machines were used to test both datasets:

Table 9. Machine specifications used for benchmark testing.

Hardware specification	Linux	MacOS X
Operating System	Red Hat 9.4	MacOS Monterey 12.5.1
CPU	144-Core Intel(R) Xeon(R) Platinum 8452Y @ 3.2 max GHz	8-Core Apple M1 Pro chip
Memory (RAM)	502 GB	16 GB

2. Test Results

The following benchmark data was generated on the two machines:

Table 10. Benchmark results, per machine for each dataset

	Total runtime	
Dataset	Linux	MacOS X
Dataset 1 (small)	7.8 min	39.1 min
Dataset 2 (large)	3 hr 28 min	*

*It is not recommended to run the large dataset on MacOS X due to resource limitations.

VI. Immune Profiler Output

NOTE: In this section, `<sampleID>` is used to represent the value of the corresponding sample ID as defined in the metadata file (see Section II.E.2, "Metadata file").

A. Output 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:

- `preprocess/` : contains QC Stats CSV files that summarize the distribution of reads across the different chain types for all samples in an analysis and, optionally, the intermediate FASTQ files created during preprocessing (`-k`/`--keep_intermediate` set to 1 or higher). This folder is covered in greater detail in Section 0 (below).

- `umi_stats/` : contains plots showing the UMI and UMI group frequencies observed per UMI group size for each chain type, to help determine user-specified UMI cutoffs for generating final reports, and a template CSV file, to be used with `cogentip report` for specifying custom UMI cutoff values.
- `split_fastq_files/` : an optional folder (`-k`/`--keep_intermediate` set to 3 or higher, refer to Section V.B.3 for more information) that contains pairs of FASTQ file for each chunk. These are used for downstream immune profiling analysis with TRUST4.

  The folder contents are further subdivided into additional nested folders:

  - `<sampleID>/` : Each sample has its own folder
    - `<chain type>` : Each sample has subfolders grouped by chain types

  Within each chain type subfolder, there is also a `umi_group_sizes_frequency.<chain type>.csv` file which contains the frequencies of UMI group sizes
- `trust4/` : contains the TRUST4 output files merged from the TRUST4 analysis chunks that are required for generating:
  - the TRUST4 final reports
  - the Immune Viewer input files

  This folder may optionally contain all TRUST4 intermediate files merged from the TRUST4 analysis chunks (`-k` 4) and all TRUST4 analysis chunk output files (`-k` ≥2), including FASTQ files created during the preprocessing step.

  Additional information about TRUST4-generated files can be found on the TRUST4 GitHub page at: https://github.com/liulab-dfci/TRUST4.
- `reports/` : This folder summarizes the TRUST4 results and contains input files for use with Cogent NGS Immune Viewer. The reports are comma-separated value (CSV) or tab-separated value (TSV) files that can be viewed in a spreadsheet program. This folder is covered in greater detail in Section VI.C (below).
- A `work/` directory will be created in the directory that the software is run from. To see the directory where `work/` will be created, run:

  ```
  pwd
  ```

  The `work/` directory contains the conda-based installations of the Cogent NGS Immune Profiler dependencies (`work/conda/`) as well as the temporary files created during an immune profiling analysis. When `cogentip` and `cogentip report` finish a successful analysis run, these temporary files are removed. When a run fails or is interrupted, the intermediate files remain in the `work/` directory to allow users to attempt to finish the analysis using the `--resume` parameter (Sections V.B.3. and V.C.3, respectively).

  The `work/` directory can be deleted if it begins to take up too much space. If the software is run from a new location, a new `work/` directory will be created there, and all the software dependencies will be reinstalled.

B. The preprocess Folder

The `preprocess/` folder is generated to contain the results of the first step of the Immune Profiler workflow, which separates reads in original sample-level FASTQs into different chain-specific FASTQs. This folder contains two QC Stats CSV files that summarize the distribution of reads (number of reads and read %) across the different chain types for all samples of an analysis. They also summarize the distribution of reads that were less than 30 nucleotides in length (short), classified as undetermined (undetermined), or failed linker-based correction (flc).

`sample_QC_stats.csv`

`sample_QC_stats.airr.csv`: Has the same content as `sample_QC_stats.csv`, but with AIRR (Adaptive Immune Receptor Repertoire)-compliant column heads and adds an organism_id column

NOTE: By default, preprocessed FASTQ files are not saved. If the option to keep intermediate files (`-k`/`--keep_intermediate`) is set to 1 or higher, preprocessed FASTQ files will be retained as well as sample-specific QC Stats CSV files.

- FASTQs can be created for each of the chain types. If a chain type is not present in the data, the corresponding files will not be created.
  - BCRv1 `<chain type>` : IgG, IgM, IgK, and IgL
  - BCRv2 `<chain type>` : IgA, IgD, IgE, IgG, IgM, IgK, and IgL
  - TCRv1 `<chain type>` : TRA and TRB
  - TCRv2 `<chain type>` : TRA and TRB

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

  `<sampleID>_undetermined_R1.fastq.gz`
  `<sampleID>_undetermined_R2.fastq.gz`
- 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.gz`
  `<sampleID>_short_R2.fastq.gz`
- If linker-based correction is turned on, an additional FASTQ pair is created to store reads that failed to correct:

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

C. The reports Folder

Immune Profiler summarizes major statistics collected by the workflow steps (Section V) and merges the results into comma-separated value (CSV) files and tab-separated value (TSV) files that can be viewed in a spreadsheet program. These files are written to and can be found in the `reports/` folder. Example output files for each receptor type are included in the mini dataset sample output in `$PROFILER_HOME/test/test_output/` (see Appendix A for more information about these files).

The final report (`reports/<sampleID>/<chain type>/<sampleID>_<chain type>.UMI_<umi cutoff>.immune_viewer_report.tsv`) is a Cogent NGS Immune Viewer-compatible, AIRR-compliant report with detected clonotypes after UMI collapse that are supported by UMI groups with a minimum size of the UMI cutoff (`-u`) value. Clonotypes missing an annotated V gene and/or J gene are filtered out. A more detailed description of the contents of this file is shown in Table 11.

Table 11. The column names and descriptions for the `<sampleID>_<chain type>.UMI_<umi cutoff>.immune_viewer_report.tsv` reports.

Column name	Description
organism_id	Value will be 'human' or 'mouse'
sample_processing_id	Value of `<sampleID>` from the metadata file
read_count	Number of UMIs (or reads, if not using UMIs) with which a particular clonotype is identified.
fraction	Fraction of read count over total reads.
clonal_sequence	Entire clonotype sequence detected
clonal_sequence_quality	N/A (Legacy value from CogentIP v1.x, not used in CogentIP v2.x)
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: • **BCRv2:** IgG, IgM, IgK, IgL, IgA, IgD, or IgE • **BCRv1:** IgG, 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`).

VII. Cogent NGS Immune Viewer

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

```
*immune_viewer_report.csv
```

files generated by Immune Profiler from the `reports/<sampleID>/<RECEPTOR>` folders (Section VI.C, above) 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 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).

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:

```
$PROFILER_HOME/test/test_input/
```

These should be used both to verify Immune Profiler is installed correctly (Section IV.F, "Conduct Test Run with the Mini Datasets") and to familiarize yourself with the operative steps to use the software.

A. Sample Input Data

Four sets of sample data corresponding to data from the following Takara Bio immune profiling kits are provided in the `test/test_input/` directory:

- SMART-Seq Human BCR (with UMIs) (Cat. Nos. 634776, 634777 & 634778)
- SMART-Seq Human TCR (with UMIs) (Cat. Nos. 634779, 634780 & 634781)
- SMART-Seq Mouse TCR (with UMIs) (Cat. Nos. 634814, 634815 & 634816)
- SMART-Seq Mouse BCR (with UMIs) (Cat. Nos. 634351, 634352 & 634353)

The mini test datasets each include a pair of FASTQ files for each sample and a metadata file that maps sample names to their corresponding FASTQ files. Refer to Figure 9 for a visual representation of the folder contents for the human BCRv2 and human TCRv2 datasets and to Table 12 for information about the numbers of samples, numbers of reads, and read length for each of the mini datasets.

```
test_input/human/
├── hBCRv2_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
│   ├── S4_R1.fastq.gz
│   ├── S4_R2.fastq.gz
│   ├── S5_R1.fastq.gz
│   ├── S5_R2.fastq.gz
│   ├── S6_R1.fastq.gz
│   ├── S6_R2.fastq.gz
│   └── hBCRv2_mini_meta.csv
└── hTCRv2_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
    └── hTCRv2_mini_meta.csv
```

Figure 9. Folder structure and files found in `test_input/`.

Table 12. Statistics of the mini dataset files.

Dataset name	# of samples	Read-pairs per sample	Read length (bp)
hBCRv2_mini	6	3x 20,000 3x 4,000	Paired-end 300
hTCRv2_mini	3	1x 2,000 1x 2,500 1x 3,000	Paired-end 151
mBCRv2_mini	2	2x 10,000	Paired-end 150
mTCRv2_mini	3	1x 6,000 1x 8,000 1x 10,000	Paired-end 301

B. Sample Results Data

Four sets of sample result files are provided in the `test/test_output/` folder corresponding to the four sets of input data in Section A. Refer to Figure 10 for a visual representation of the folder contents for the human BCRv2 and human TCRv2 results.

```
test_output/human/hBCRv2_mini_results/
├── preprocess/
│   ├── sample_QC_stats.airr.csv
│   └── sample_QC_stats.csv
├── reports/
│   ├── S1/
│   │   ├── IGA/
│   │   │   ├── S1_IGA.UMI_1.immune_viewer_filtered.csv
│   │   │   ├── S1_IGA.UMI_1.immune_viewer_report.csv
│   │   │   ├── TRUST4_airr.UMI_1.tsv
│   │   │   ├── TRUST4_barcode_airr.tsv
│   │   │   ├── TRUST4_barcode_report.tsv
│   │   │   └── TRUST4_report.UMI_1.tsv
│   │   └── ...
│   └── ...
├── trust4/
│   ├── S1/
│   │   ├── IGA/
│   │   │   ├── TRUST4_airr_align.tsv
│   │   │   ├── TRUST4_annot.fa
│   │   │   └── TRUST4_cdr3.out
│   │   └── ...
│   └── ...
└── umi_stats/
    ├── umi_cutoffs.template.csv
    ├── umi_group_sizes_frequency.S1.png
    └── ...
```

```
test_output/human/hTCRv2_mini_results/
├── preprocess/
│   ├── sample_QC_stats.airr.csv
│   └── sample_QC_stats.csv
├── reports/
│   ├── S4/
│   │   ├── TRA/
│   │   │   ├── S4_TRA.UMI_1.immune_viewer_filtered.csv
│   │   │   ├── S4_TRA.UMI_1.immune_viewer_report.csv
│   │   │   ├── TRUST4_airr.UMI_1.tsv
│   │   │   ├── TRUST4_barcode_airr.tsv
│   │   │   ├── TRUST4_barcode_report.tsv
│   │   │   └── TRUST4_report.UMI_1.tsv
│   │   └── ...
│   └── ...
├── trust4/
│   ├── S4/
│   │   ├── TRA/
│   │   │   ├── TRUST4_airr_align.tsv
│   │   │   ├── TRUST4_annot.fa
│   │   │   └── TRUST4_cdr3.out
│   │   ├── TRB/
│   │   │   ├── TRUST4_airr_align.tsv
│   │   │   ├── TRUST4_annot.fa
│   │   │   └── TRUST4_cdr3.out
│   │   └── ...
│   └── ...
└── umi_stats/
    ├── umi_cutoffs.template.csv
    ├── umi_group_sizes_frequency.S4.png
    └── ...
```

Figure 10. Folder structure and files found in the `test_output/` folders. **(Left)** `human/hBCRv2`. **(Right)** `human/hTCRv2`.

- **Files**
  - `preprocess/sample_QC_stats.csv`: quality control statistics for all samples in the analysis
  - `preprocess/sample_QC_stats.airr.csv`: quality control statistics in an AIRR-compliant format for all samples in the analysis
  - `umi_stats/umi_group_sizes_frequency.<sampleID>.png`: plots showing the UMI and UMI group frequencies observed per UMI group size for each chain type to help determine user-specified UMI cutoffs for generating final reports

- umi_stats/umi_cutoffs.template.csv: template CSV file to be used with cogentip report for specifying custom UMI cutoff values for each sample and chain type
- trust4/<sampleID>/<chain type>/TRUST4_airr_align.tsv: intermediate TRUST4-generated file (with alignment information of each consensus assembly to the V, D, J, and C genes) that is required to generate final reports
- trust4/<sampleID>/<chain type>/TRUST4_cdr3.out: intermediate TRUST4-generated file (with CDR1, CDR2, and CDR3 and V, D, J, and C gene information for each consensus assembly) that is required to generate final reports
- trust4/<sampleID>/<chain type>/TRUST4_annot.fa: intermediate TRUST4-generated file (with each consensus assembly in FASTA format annotated with CDR1, CDR2, and CDR3 and V, D, J, and C gene information and their locations within the consensus assembly) that is required to generate final reports
- reports/<sampleID>/<chain type>/TRUST4_barcode_report.tsv: TRUST4-generated report with detected clonotypes and their associated UMIs
- reports/<sampleID>/<chain type>/TRUST4_barcode_airr.tsv: AIRR-compliant TRUST4-generated report with detected clonotypes and their associated UMIs
- reports/<sampleID>/<chain type>/TRUST4_report.UMI_<umi cutoff>.tsv: TRUST4-generated report with detected clonotypes after UMI collapse that are supported by UMI groups with a size of at least <umi cutoff>
- reports/<sampleID>/<chain type>/TRUST4_airr.UMI_<umi cutoff>.tsv: AIRR-compliant TRUST4-generated report with detected clonotypes after UMI collapse that are supported by UMI groups with a size of at least <umi cutoff>
- reports/<sampleID>/<chain type>/<sampleID>_<chain type>.UMI_<umi cutoff>.immune_viewer_report.tsv: Cogent NGS Immune Viewer-compatible, AIRR-compliant report with detected clonotypes after UMI collapse that are supported by UMI groups with a size of at least <umi cutoff>. Clonotypes missing an annotated V gene and/or J gene are filtered out.
- reports/<sampleID>/<chain type>/<sampleID>_<chain type>.UMI_<umi cutoff>.immune_viewer_filtered.tsv: clonotypes that have been filtered out for missing an annotated V gene and/or J gene while preparing the report that can be used as input to Cogent NGS Immune Viewer

The following files 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

- TRUST4_airr.UMI_<umi_cutoff>.tsv
- TRUST4_barcode_airr.tsv
- <sampleID>_<chain type>.UMI_<umi_cutoff>.immune_viewer_report.csv
- <sampleID>_<chain type>.UMI_<umi_cutoff>.immune_viewer_filtered.csv

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.

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