# BD Rhapsody™

## Sequence Analysis Pipeline

### User's Guide

23-24580(01)
2024-02

For Research Use Only

## Copyrights

The information in this guide is subject to change without notice. BD reserves the right to change its products and services at any time. Although this guide has been prepared with every precaution to ensure accuracy, BD assumes no liability for any errors or omissions, nor for any damages resulting from the application or use of this information. BD welcomes customer input on corrections and suggestions for improvement.

## Patents and trademarks

For US patents that may apply, see bd.com/patents.

## Regulatory information

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

## History

| Revision | Date | Change made |
|---|---|---|
| 23-24580(01) | 2024-02 | Initial release to support Rhapsody™ Sequence Analysis Pipeline v2.2. This user guide integrates and supersedes the *Single-Cell Multiomics Bioinformatics Handbook*, 23-21713, and *Single-Cell Multiomics Analysis Setup User Guide*, 23-21333. |

# Contents

1. Introduction 5

2. Setup and Running Guide 7

Input Files and Parameters 8
Seven Bridges Setup 14
Local Server Setup 20

3. Pipeline Steps and Algorithms 27

Read Quality Filter 29
Annotate R1 Cell Label and UMI 31
Align and Annotate R2 Reads 33
Valid Read Pairs 34
Molecules and Error Correction 35
Determine Putative Cells 41
Sample Tag Analysis 52
Generate Expression Matrix Data Tables 55
Annotate BAM 56
TCR and BCR Analysis 57
ATAC-Seq Analysis 61

4. Output Files 65

Metrics Summary CSV 68
Pipeline Report HTML 73
Cell-by-Feature Data Tables 80
BAM and BAM Index 81
Single-Cell Analysis Tool Inputs 82
Bioproduct Statistics 83
Sample Tag Metrics 85
Sample Tag Calls 86
VDJ Metrics 88
VDJ Per Cell 91
VDJ Per Cell Uncorrected 92
VDJ Dominant Contigs AIRR 93
VDJ Unfiltered Contigs AIRR 96
Protein Aggregates Experimental 98
Immune Cell Classification Result 99

Dimensionality Reduction Coordinates ........ 100
ATAC Fragments and Fragments Index ........ 101
ATAC Transposase Sites and Index file ........ 102
ATAC Peaks and Peaks Index ........ 103
ATAC Cell-by-Peak Data Tables ........ 104
ATAC Metrics ........ 105

**5. Troubleshooting Guide 111**

Access Log Files ........ 112
Output Metrics and Associated Problems ........ 114
Skim Sequencing for Assessing Library Quality ........ 117

**6. Additional Resources 119**

Extra Utilities ........ 120
Scripts for BD Rhapsody™ Data ........ 124
Pipeline Install Bundle (Docker-Free) ........ 125

**7. References 127**

**8. Release Notes 131**

# 1

# Introduction

This guide provides detailed instructions on how to set up and run the BD Rhapsody™ Sequence Analysis Pipeline on the Seven Bridges Genomics platform or on a local server installation.

This pipeline performs analysis of single-cell multiomic sequence read (FASTQ) data. The supported sequencing libraries are those generated by the BD Rhapsody™ assay kits, including: Whole Transcriptome mRNA, Targeted mRNA, AbSeq Antibody-Oligonucleotides, Single-Cell Multiplexing, TCR/BCR, and ATAC-Seq.

Included is a comprehensive reference to help you understand the pipeline's step-by-step process, the analysis algorithms, the output files, and the metrics. For any problems, refer to the included Troubleshooting guide, or reach out to the support team.

***- The BD Single-Cell Multiomics team***

# 2

# Setup and Running Guide

Whether analysis is performed on the Seven Bridges Genomics platform or on a local server, sequencing analysis uses the same BD Rhapsody™ Sequence Analysis Pipeline. The necessary input files and optional parameters are common to both run modes. During the execution of the pipeline, sequencing analysis processes sequence read files to generate molecular counts per bioproduct per cell, metrics, and possibly other files applicable to the assay types performed in the experiment.

This section includes detailed instructions on how to set up and run the BD Rhapsody™ Sequence Analysis Pipeline. The main topics are:

- Input Files and Parameters (page 8)
- Seven Bridges Setup (page 14)
- Local Server Setup (page 20)

# Input Files and Parameters

The file inputs required to run this pipeline are FASTQ read files, and reference files, the identity of which will depend on the assay type and species of cells in your experiment.

There are also many optional paramerters that are used to specify the inclusion of particular assay types, or adjust how the pipeline works.

## FASTQ Files

### Read 1, Read 2, and Index Read 2 sequencing files

For either the Seven Bridges Genomics platform or local pipeline installation, first obtain sequence FASTQ files: Read 1 and Read 2. For ATAC-Seq assays, also obtain Index Read 2. Although the FASTQ filenames can have any format, we recommend the following:

- Include `R1`, `R2`, or `I2`
- The base name should be the same for R1, R2 and I2
- Convert uncompressed files to .gz format

**Example library FASTQ files for WTA assay:**

- WTALibrary1_S1_L001_R1_001.fastq.gz
- WTALibrary1_S1_L001_R2_001.fastq.gz

**Example library FASTQ files for ATAC-Seq assay:**

- ATACLibrary_S2_L001_R1_001.fastq.gz
- ATACLibrary_S2_L001_R2_001.fastq.gz
- ATACLibrary_S2_L001_I2_001.fastq.gz

**Do not use special characters or spaces in the filenames, or the analysis might fail. Use only letters, numbers, underscores, or hyphens.**

**Note:** If you are downloading the files from BaseSpace, follow these steps:

1. Choose the run to download in BaseSpace
2. Click the Download icon on the main screen
3. If necessary, install the BaseSpace downloading application
4. Click **Select all fastq files for this run**
5. Download the files. This might take several minutes

For more information, go to help.basespace.illumina.com.

# Reference Files

## Introduction

For targeted mRNA assays, FASTA reference files are used to store the sequences of gene targets.

For whole transcriptome assays (WTA), the reference files archive is a compressed tarball that contains the STAR (page 129) index files and the GTF transcriptome annotation corresponding to the species of cells used in the BD® WTA experiment.

For ATAC-Seq or Multiomic ATAC-Seq (WTA+ATAC-Seq) assays, the reference files archive is a compressed tarball that contains all the contents as described for a WTA assay, an additional index for bwa-mem2 (page 127), and a text file containing the mitochondrial contig names.

The AbSeq Reference is a FASTA file for BD® AbSeq Ab-Oligos used in a BD Rhapsody™ experiment.

If additional transgene sequences are used in the experiment, an additional FASTA file containing the sequences can be used as the Supplemental Reference.

## Obtaining pre-designed targeted mRNA panels, WTA, or Multiomic WTA+ATAC-Seq reference files

Obtain the targeted FASTA references from the Seven Bridges demo project, or by contacting BD Biosciences customer support at scomix@bdscomix.bd.com.

For WTA assays, obtain a pre-built reference genome archive file for human or mouse from the Seven Bridges demo project, or by downloading from the following link: bd-rhapsody-public.s3-website-us-east-1.amazonaws.com/Rhapsody-WTA/

For ATAC-Seq and Multiomic WTA+ATAC-Seq assays, obtain a pre-built reference genome archive file for human or mouse from the Seven Bridges demo project, or by downloading from the following link: bd-rhapsody-public.s3-website-us-east-1.amazonaws.com/Rhapsody-WTA-ATAC/

## Pre-built WTA reference gene biotypes

The GTF file in the pre-built WTA reference archive has been preprocessed to contain only the following gene types: `protein_coding, lncRNA, lincRNA, antisense, IG_LV_gene, IG_V_gene, IG_V_pseudogene, IG_D_gene, IG_J_gene, IG_J_ pseudogene, IG_C_gene, IG_C_pseudogene, TR_V_gene, TR_V_pseudogene, TR_D_gene, TR_J_gene, TR_J_pseudogene, and TR_C_gene`

## Designing custom Targeted mRNA panels

By providing a list of genes to BD Biosciences customer support, we can design custom mRNA targeted panels. Contact BD Biosciences customer support at scomix@bdscomix.bd.com.

## AbSeq reference files

If your experiment contains BD® AbSeq Ab-Oligos, you are required to have an AbSeq reference file. To prepare the AbSeq reference file, you can use the BD AbSeq Panel Generator (abseq-ref-gen.genomics.bd.com) or perform the following instructions.

1. Download the FASTA file containing all of the BD Ab-Oligo (AbO) sequence. Go to bd-rhapsody-public.s3-website-us-east-1.amazonaws.com/AbSeq-references/BDAbSeq_allReference_latest.fasta.

2. Use a text editor such as Microsoft® Notepad or TextEdit to delete the sequence header and sequence pairs that will not be used in the experiment.

   **Do not use a word processor such as Microsoft® Word, which can add unintended special characters to the file.**

3. Ensure that the AbSeq reference file follows these rules:

   - File extension is `.fa` or `.fasta`

   - Two line fasta format. Format example:

     ```
     >CD103|ITGAE|AHS0001|pAbO
     AAATAGTATCGAGCGTAGTTAAGTTGCGTAGCCGTT
     >CD161:DX12|KLRB1|AHS0002|pAbO
     GTTATGGTTGTCGGTAGAGTATCGTGTTGCGTTAGT
     ```

     **Note:** BD Biosciences uses this format for its sequence header: `<AntibodyName>|<GeneSymbol>|<SeqID>|pAbO`.

## Building a custom WTA only or Multiomic WTA+ATAC-Seq reference archive

The WTA reference archive is a tar.gz file with the following internal structure:

```
BD_Rhapsody_Reference_Files/ # top level folder
   star_index/ # sub-folder containing STAR index
      [files created with STAR --runMode genomeGenerate]
   GTF for gene-transcript-annotation e.g. "gencode.v43.primary_assembly.annotation.gtf"
```

The WTA+ATAC-Seq reference archive is a tar.gz file with the following internal structure:

```
BD_Rhapsody_Reference_Files/ # top level folder
   star_index/ # sub-folder containing STAR index
      [files created with STAR --runMode genomeGenerate]
   GTF for gene-transcript-annotation e.g. "gencode.v43.primary_assembly.annotation.gtf"
   mitochondrial_contigs.txt # mitochondrial contigs in the reference genome - one contig
name per line. e.g. chrMT or chrM, etc.
   bwa-mem2_index/ # sub-folder containing bwa-mem2 index
       [files created with bwa-mem2 index]
```

The same docker image used for running the BD Rhapsody™ Sequence Analysis Pipeline can be used for generating a WTA-only or WTA+ATAC-Seq reference archive with the following steps:

1. Goto bitbucket.org/CRSwDev/cwl and download the Extra_Utilities file: `make_rhap_reference_<version>.cwl`

2. Gather a matching genome-sequence set in FASTA format and GTF with gene, transcript, and exon annotations, for example, from gencodegenes.org.

3. Run `cwl-runner` like the following example :

   ```
   cwl-runner make_rhap_reference_2.0.cwl --Genome_fasta GRCh38.primary_
   assembly.genome.fa --Gtf gencode.v43.primary_assembly.annotation.gtf --
   Archive_prefix testrefhuman43
   ```

   The resulting `testrefhuman43.tar.gz` file can be used for the Reference_Archive input of the BD Rhapsody™ Sequence Analysis Pipeline. By default the combined WTA+ATAC-Seq reference is created.

   To create a WTA only index pass the flag --WTA_only, i.e. : `cwl-runner make_rhap_reference_2.0.cwl --Genome_fasta GRCh38.primary_assembly.genome.fa --Gtf gencode.v43.primary_assembly.annotation.gtf --Archive_prefix testrefhuman43 --WTA_only`

## Pipeline Parameters

The following table describes both the input files and optional parameters that can be set when running the Sequence Analysis Pipeline. These parameters are applicable to running the pipeline on either Seven Bridges (where they are set with the graphical user interface), or on a local server (where they are set using the input specification YML file).

### Required and optional inputs and parameters

| Input field | Input | Required? |
|---|---|---|
| ATAC_Predefined_Peak_Regions | File input: An optional BED file (such as the ATAC-Seq peaks file output by the Rhapsody pipeline) containing pre-established chromatin accessibility peak regions for generating the ATAC-Seq cell-by-peak matrix. Useful if a direct comparison of chromatin accessibility between two or more ATAC-Seq samples is desired. | Optional for ATAC-Seq assay |
| AbSeq_Reference | File input: FASTA AbSeq reference file as described in the Input Files and Parameters (page 8) section. Ensure that the AbSeq reference file contains only the BD AbSeq Ab-Oligos that were used in the experiment. | Optional |
| Cell_Calling_ATAC_Algorithm | Default: Basic. Specify the putative cell calling algorithm for ATAC-Seq: Basic, Refined. | Optional |
| Cell_Calling_Bioproduct_Algorithm | Default: Basic. Specify the putative cell calling algorithm for bioproducts: Basic, Refined | Optional |
| Cell_Calling_Data | Default: mRNA. Specify the data to be used for putative cell calling: mRNA, AbSeq, ATAC, mRNA_and_ATAC. | Optional |

| Input field | Input | Required? |
|---|---|---|
| Exact_Cell_Count | Set a specific number (>=1) of cells as putative, based on those with the highest error-corrected read count. | Optional |
| Exclude_Intronic_Reads | Default: False. By default, reads aligned to exons and introns are considered and represented in molecule counts. Including intronic reads may increase sensitivity, resulting in an increase in molecule counts and the number of genes per cell for both cellular and nuclei samples. Intronic reads may indicate unspliced mRNAs and are also useful, for example, in the study of nuclei and RNA velocity. When set to True, intronic reads will be excluded. | Optional |
| Expected_Cell_Count | Guide the basic putative cell calling algorithm by providing an estimate of the number of cells expected. Usually this can be the number of cells loaded into the BD Rhapsody™ cartridge. | Optional |
| Generate_Bam | Default: False. A Bam read alignment file contains reads from all the input libraries, but creating it can consume a lot of compute and disk resources. By setting this field to True, the Bam file will be created. | Optional |
| Reads | File input: R1 reads and R2 reads. Ensure to include all FASTQ sequencing data from the experiment, including R1 and R2 files for the targeted or WTA RNA library, and, if applicable, the Sample Tag, TCR, BCR, and BD® AbSeq libraries. | Required for applicable libraries |
| Reads_ATAC | File input: R1, R2 and I2 reads. Ensure to include all FASTQ sequencing data from the experiment, including R1, R2 and I2 files for the ATAC-Seq library. | Required for ATAC-Seq libraries |
| Reference_Archive (WTA or WTA+ATAC-Seq) | File input: A TAR.GZ file that includes a STAR (and possibly a bwa-mem2) indexed reference genome file, along with a GTF gene annotation file. | Yes |
| Run_Name | Specify a run name to be used as the base output filename. Use only letters, numbers, hyphens, or underscores. If any other special characters are included, they will be corrected to hyphens. | Optional |
| Sample_Tags_Version | For a multiplexed samples run only. Specifies the Sample Tags used: human (hs), mouse (mm), or flex. | Required for multiplexed samples |
| Supplemental_Reference | File input: This is a FASTA file that contains additional transgene sequences. | Optional |

| Input field | Input | Required? |
|---|---|---|
| Tag_Names | For a multiplexed samples run only. Associate a name with each Sample Tag, which will appear in the output files. Within square brackets, enter a comma-separated list of Sample Tag numbers and associated names. For each sample, use the following format, using a hyphen—**no spaces or forward slashes allowed:**<br><br>**Sample Tag number-sample name**<br>Example: Tag_Names: [3-Ramos, 4-BT549] | Optional for multiplexed samples |
| Targeted_Reference (Targeted only) | File input: FASTA file containing the sequences amplified by the primers of the Targeted assay. This can be a pre-designed, supplemental, or custom panel. Ensure that the reference matches the species and panel used for the experiment. Otherwise, read mapping will not be correctly aligned. | Yes |
| VDJ_Version | For experiments with VDJ libraries. Specify the species or chain types or both. Species-only selection will include both BCR and TCR. Options:<br>human<br>mouse<br>humanBCR<br>humanTCR<br>mouseBCR<br>mouseTCR | Required for TCR/BCR assay |

# Seven Bridges Setup

### Steps

1. Create a Seven Bridges Account (page 14)
2. Create a New Project (page 15)
3. Upload FASTQ Files (page 16)
4. Upload Reference Files (page 16)
5. Import the Pipeline App (page 17)
6. Create Task and Run the Pipeline (page 17)
7. Download the Output (page 19)

## Create a Seven Bridges Account

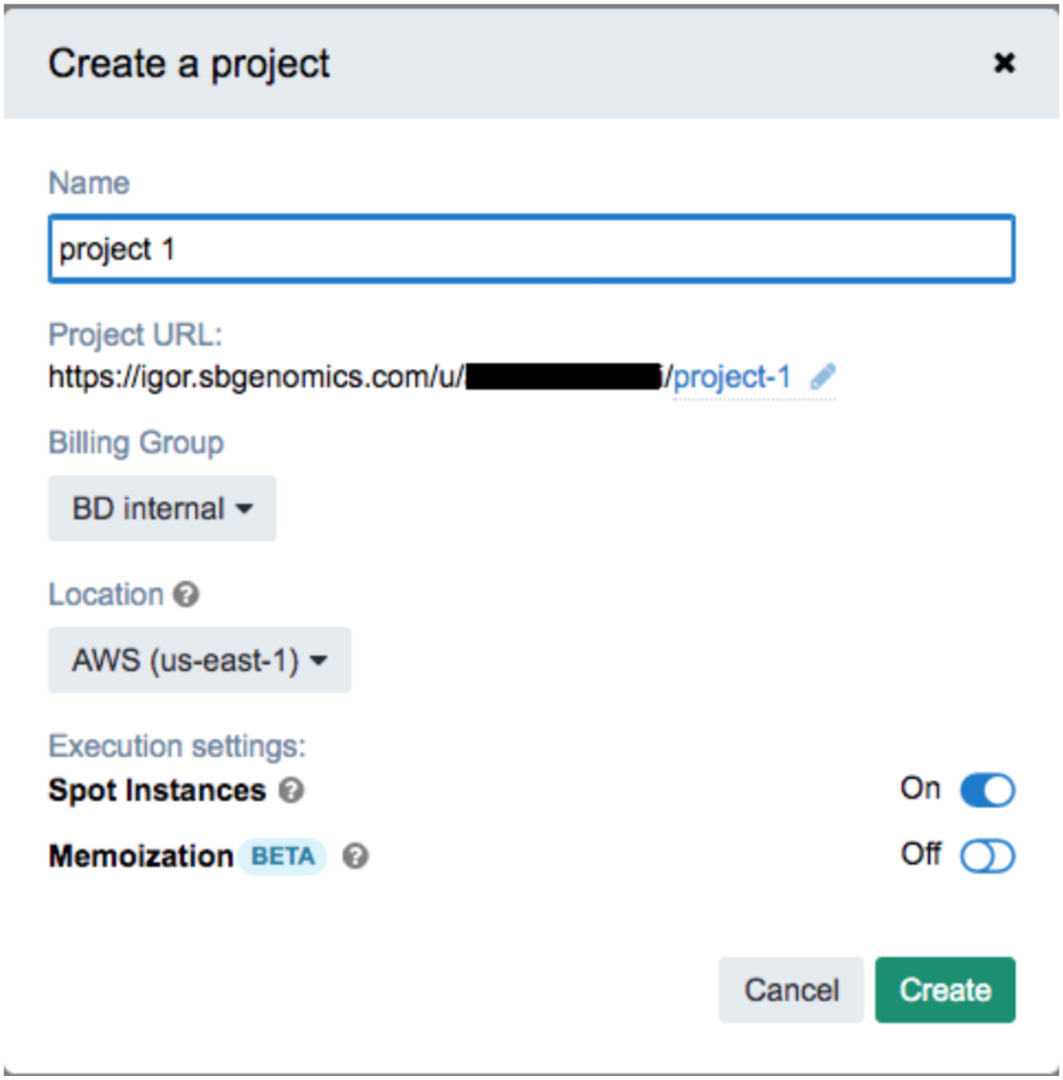
Create an account only if you will analyze sequencing data on the Seven Bridges Genomics platform.

1. Go to sevenbridges.com/bdgenomics/.
2. Click **Request Access.** In the request access window, enter your email address so that you can receive an email invitation to the Seven Bridges Genomics platform within 24 hours.
3. Click the link in the email invitation, and complete the registration. Seven Bridges Genomics displays the dashboard with the demo projects.

## Create a New Project

### Procedure

1. At the top of the dashboard, click **Projects > Create a project**:



Create a project

Name: project 1

Project URL: https://igor.sbgenomics.com/u/[redacted]/project-1

Billing Group: BD internal

Location: AWS (us-east-1)

Execution settings:

Spot Instances: On

Memoization BETA: Off

**Note:** To enable automatic reuse of intermediate files in a rerun, turn **Memoization** on.

2. On the Create a project dialog, enter the project name, and edit the project URL if necessary.

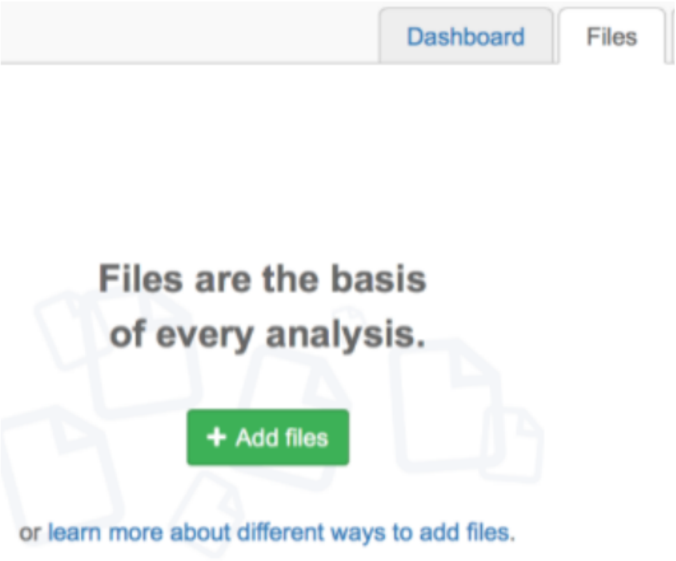
3. Click **Create**. Seven Bridges Genomics displays the new project dashboard.

4. To change the retention period of intermediate files, click **Settings** in the top right corner. Enter 120 in the Retention period box to specify the number of hours for retention, and click **Save**. This may help to troubleshoot a pipeline run if there are any problems.

# Upload FASTQ Files

## Procedure

1. On the project dashboard, click the **Files** tab, and then click **+ Add files**:



Files are the basis of every analysis.

or learn more about different ways to add files.

2. In the top menu, select the source of the files, such as **Public files, Projects, Your Computer,** or **FTP/HTTP**. Seven Bridges Genomics displays instructions on uploading the files. Follow the Seven Bridges Genomics instructions to import your files.

3. After upload, the files are listed on the Files tab.

# Upload Reference Files

## Upload custom mRNA, AbSeq, ATAC-Seq or supplemental reference files

1. On the project dashboard, click the **Files** tab, and then click **+ Add files**.

2. In the top menu, select the source of the files, such as **Public files, Projects, Your Computer,** or **FTP/HTTP**. Seven Bridges Genomics displays instructions on uploading the files. Follow the Seven Bridges Genomics instructions to import your files.

3. After upload, the files are listed on the Files tab.

## Copy reference files from an existing Demo project

1. On the Files tab of the project dashboard, click **+ Add files**.

2. Click **Projects**, and then click **Demo Project** in the left panel.

3. Do one of the following:

   - For Targeted assays: Locate the appropriate FASTA file for your experiment, and click **Copy**.
   - For AbSeq assays: Locate the appropriate FASTA file for your experiment, and click **Copy**.
   - For WTA only and Multiomic WTA+ATAC-Seq assays: Locate the appropriate reference genome archive for your experiment, and click **Copy**.

## Import the Pipeline App

1. On the project dashboard, click the **Apps** tab, and then click **+ Add app**.
2. Click **Public Apps**, and then enter "Rhapsody" to find the appropriate pipeline, called BD Rhapsody™ Sequence Analysis Pipeline. Or, copy the app from the Demo project.
3. Click **Copy** on the app window, select the project in the dropdown menu, and then click **Copy** again.
4. Navigate to the Apps tab to confirm that the workflow app was copied to the project.

## Create Task and Run the Pipeline

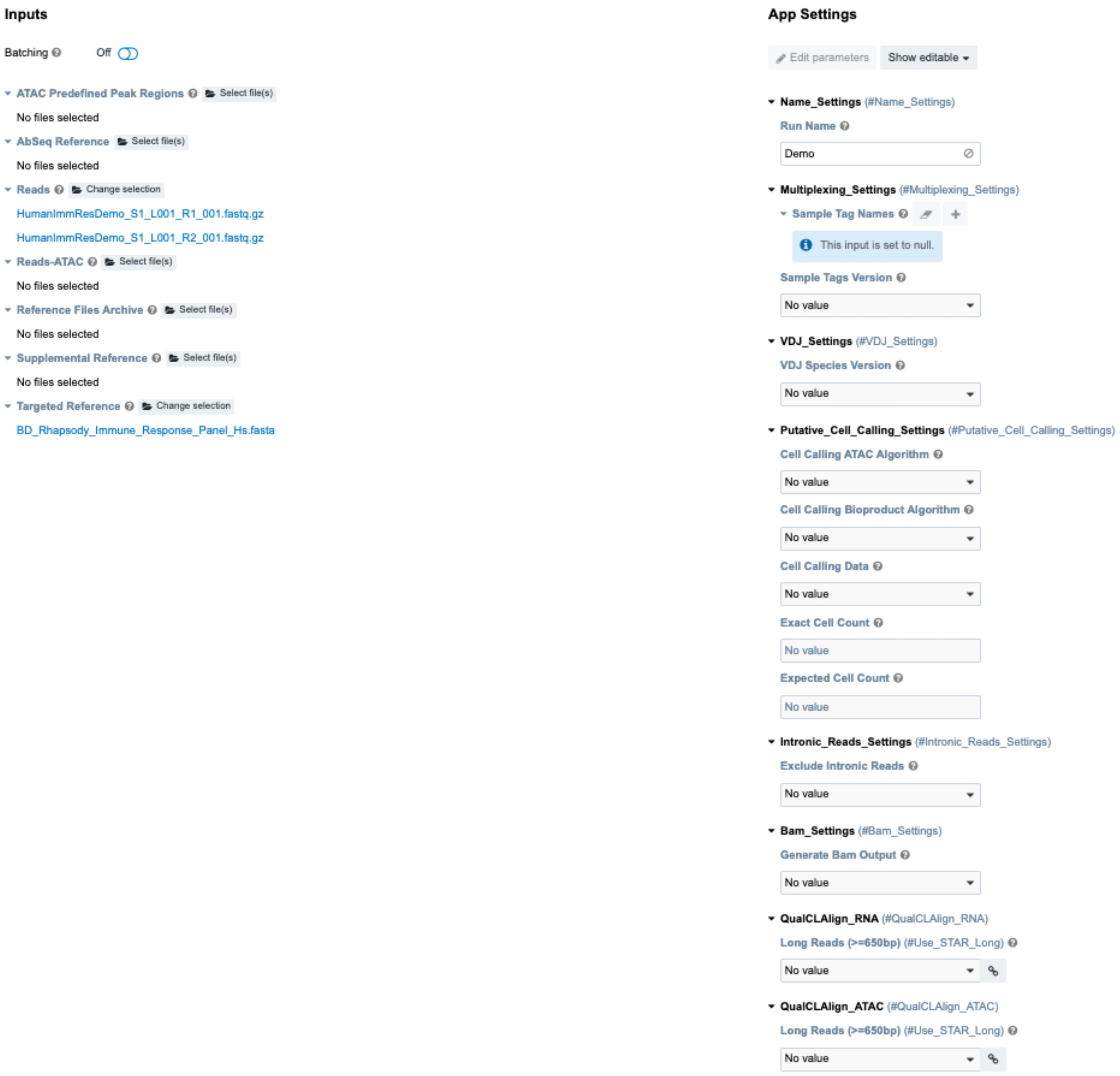
### Procedure

1. Click the **Apps** tab to view the apps.

   **Note:** If the app is highlighted in yellow, an update is available. Select the update link to get the latest app version.

2. By the BD Rhapsody™ Sequence Analysis Pipeline, click the green play button under Actions. The Task Inputs tab will display the Inputs and App Settings.

## Sequence Analysis Pipeline interface:



**Inputs**

Batching Off

- ATAC Predefined Peak Regions — Select file(s)
  No files selected
- AbSeq Reference — Select file(s)
  No files selected
- Reads — Change selection
  HumanImmResDemo_S1_L001_R1_001.fastq.gz
  HumanImmResDemo_S1_L001_R2_001.fastq.gz
- Reads-ATAC — Select file(s)
  No files selected
- Reference Files Archive — Select file(s)
  No files selected
- Supplemental Reference — Select file(s)
  No files selected
- Targeted Reference — Change selection
  BD_Rhapsody_Immune_Response_Panel_Hs.fasta

**App Settings**

Edit parameters | Show editable

- **Name_Settings** (#Name_Settings)
  Run Name: Demo
- **Multiplexing_Settings** (#Multiplexing_Settings)
  Sample Tag Names: This input is set to null.
  Sample Tags Version: No value
- **VDJ_Settings** (#VDJ_Settings)
  VDJ Species Version: No value
- **Putative_Cell_Calling_Settings** (#Putative_Cell_Calling_Settings)
  Cell Calling ATAC Algorithm: No value
  Cell Calling Bioproduct Algorithm: No value
  Cell Calling Data: No value
  Exact Cell Count: No value
  Expected Cell Count: No value
- **Intronic_Reads_Settings** (#Intronic_Reads_Settings)
  Exclude Intronic Reads: No value
- **Bam_Settings** (#Bam_Settings)
  Generate Bam Output: No value
- **QualCLAlign_RNA** (#QualCLAlign_RNA)
  Long Reads (>=650bp) (#Use_STAR_Long): No value
- **QualCLAlign_ATAC** (#QualCLAlign_ATAC)
  Long Reads (>=650bp) (#Use_STAR_Long): No value

Complete all required fields (which appear in red), and all desired optional App Settings (may depend on assay).

1. In the Inputs section, import your files for analysis according to these requirements:

   - For every `R1.fastq.gz` file, import the paired `R2.fastq.gz` file. For ATAC-Seq, also include the corresponding index read `I2.fastq.gz` file.

- Multiple sequencing libraries can be run together as long as they are from the same set of cells, but the files and libraries can be generated from different sequencer runs.
- Specify at least one reference for the assay type(s) and species of cells in the experiment.

2. If necessary, set other input parameters in the App Settings section. For example:

   When using a BD® Single-Cell Multiplexing Kit, be sure to select the Sample_Tags_Version (Single-Cell Multiplex Kit - Human, Mouse, or Flex) from the dropdown menu.

   See Pipeline Parameters (page 11) for details on each app setting.

3. Click Run. Seven Bridges Genomics displays the app running on the Tasks tab.
4. If you enabled email notifications, look for notification of the completed run.

## Download the Output

1. Select the project from the Projects drop-down menu to view output files.
2. Click the **Tasks** tab to view the list of tasks.
3. Click the name of the completed task to view Outputs on the right of the screen.
4. Click the output file to view it, and click **Download** to download and save the output file. To download more than one output file at a time, click the Folder icon to the right of Outputs. Click the check boxes by files to download, or click the gray check box at the top to select all files, and then click **Download**.

# Local Server Setup

### Steps

1. System and Software Installation (page 20)
2. Create the Input Specification File (page 22)
3. Run the Pipeline with CWL-runner (page 25)

## System and Software Installation

### Introduction

The software applications required for analysis have specific software tools. To ensure that these tools are always available, the analysis is run in a self-contained environment called a docker container. The docker container is obtained by “pulling” or downloading a docker image to your local computer. The docker container has all of the libraries and settings required by the pipeline to run the analysis. In the portable docker container, the analysis can be run reproducibly wherever it is deployed, whether on a local installation or the Seven Bridges Genomics platform. CWL-runner is the tool that manages docker containers to complete the pipeline run. CWL-runner uses two inputs: a CWL workflow file and a YML input specification file. The CWL workflow file describes each step in the pipeline and how each docker container should run to complete the step. The YML file tells CWL-runner where to find the pipeline inputs, such as the sequencer read files (fastqs) and reference. When the pipeline run is finished, CWL-runner obtains the final outputs in the docker containers and adds them to a designated output folder on your computer.

### Minimum system requirements

- Operating system: macOS® or Linux®. (Microsoft® Windows® is not supported)
- 8-core processor (>16-core recommended)
- RAM
  - Targeted assays: 32 GB RAM (>128 GB recommended)
  - WTA and ATAC-Seq assays: 96 GB (>192 GB recommended)
- 250 GB free disk space (>1 TB recommended)

### Software requirements

**Docker**

Install the Docker Engine. docs.docker.com/engine/install/

Ensure that docker is running by entering `docker` at the command line. The docker manual should print to the terminal screen.

**Python 3**

1. Check to see if a version of Python 3 is already installed by running at the command line:

   ```
   $ python3 --version
   ```

2. Ensure that you are using a local installation of Python and not a system version. Run:

   ```
   $ which python
   ```

This should return the path to a local installation and not to a system path (usually `/usr/bin/python`).

**Using a system installation of python might not give you sufficient permissions to install the required packages.**

3. If a version of Python 3 is not installed, download and install it from python.org/downloads.

4. Update pip before installing cwlref-runner by using the command:

   ```
   $ pip install -U pip
   ```

### CWL-runner

1. Install the package from PyPi. Enter:

   ```
   $ pip install cwlref-runner
   ```

2. Ensure that cwl-runner is in your path. Type:

   ```
   $ cwl-runner
   ```

3. If the command is not found, add the install location of the pip packages to $PATH.

   a. Find where cwl ref-runner is installed by entering:

      ```
      $ pip show cwlref-runner
      ```

   b. Add the displayed path to `$PATH.` For example:

      ```
      $ export
      PATH=$PATH:/Library/Frameworks/Python.framework/Versions/3.6/lib/pytho
      n3
      ```

   c. Restart the command line utility.

### CWL and YML files

Ensure that you are using the correct CWL files with your pipeline, or the analysis might fail.

1. Go to bitbucket.org/CRSwDev/cwl.
2. In the left pane, click **Downloads > Download Repository**. The CWL and example YML files are downloaded.
3. Unzip the archive. Each folder within the archive is named after the pipeline version it corresponds to.

### Pipeline image

1. Ensure that docker is running.

2. Download (pull) the docker image by entering:

```
$ docker pull bdgenomics/rhapsody
```

**Note:** The pull command automatically downloads the most current pipeline version. To download an earlier version, specify the version number. For example:

```
$ docker pull bdgenomics/rhapsody:v1.0
```

3. Confirm the pipeline image by entering:

```
$ docker images
```

**Note:**

- `bdgenomics/rhapsody` is displayed under the repository column.
- The pipeline version number is displayed under the tag column.

## Create the Input Specification File

### Procedure

An example input specification file `pipeline_inputs_template.yml` is found in the CWL folder for the specific pipeline version. It is generally recommended to start with this file and modify it with your specific needs.

1. Obtain the FASTQ files. See FASTQ Files (page 8).

2. Obtain the targeted mRNA reference FASTA file or the WTA or WTA+ATAC-Seq Reference Files Archive file. See Reference Files (page 9).

3. If your experiment contains BD® AbSeq Ab-Oligos, obtain the AbSeq Reference file. See Reference Files (page 9).

4. Specify the file paths in the YML file for Reads and Reference with the exact input field listed in the Pipeline Parameters (page 11) table.

   - The required input fields for Targeted assays are: Reads and Targeted_Reference.

   - The required input fields for WTA assays are: Reads and Reference_Archive.

   - The required input fields for ATAC-Seq assays are: Reads_ATAC and Reference_Archive.

   - The required input fields for Multiomic WTA+ATAC-Seq assays are: Reads, Reads_ATAC and Reference_Archive.

   - The required input fields for AbSeq-only assays are: Reads and AbSeq_Reference.

5. Set any other optional parameters applicable to your assay, for instance, Sample Multiplexing, VDJ version, or cell calling parameters. See possible parameters in the `pipeline_inputs_template.yml` file, or Pipeline Parameters (page 11).

6. Save the modified template YML file

Example YML input specification files:

**Targeted assay:**

```
#!/usr/bin/env cwl-runner
cwl:tool: rhapsody
Reads:
 - class: File
   location: "path/to/mySample_R1_.fastq.gz"
 - class: File
   location: "path/to/mySample_R2_.fastq.gz"
Targeted_Reference:
 - class: File
   location: "path/to/BD_Rhapsody_Immune_Response_Panel_Hs.fasta"
```

**WTA assay with AbSeq:**

```
#!/usr/bin/env cwl-runner
cwl:tool: rhapsody
Reads:
 - class: File
   location: "path/to/WTALibrary_R1_.fastq.gz"
 - class: File
   location: "path/to/WTALibrary_R2_.fastq.gz"
 - class: File
   location: "path/to/AbSeqLibrary_R1_.fastq.gz"
 - class: File
   location: "path/to/AbSeqLibrary_R2_.fastq.gz"
Reference_Archive:
  class: File
  location: "path/to/RhapRef_Human_WTA_2023-02.tar.gz"
AbSeq_Reference:
 - class: File
   location: "path/to/AbSeq_reference.fasta"
```

**WTA assay with Sample multiplexing and TCR/BCR(VDJ) analysis:**

```
#!/usr/bin/env cwl-runner
cwl:tool: rhapsody
Reads:
 - class: File
   location: "path/to/WTALibrary_R1_.fastq.gz"
 - class: File
   location: "path/to/WTALibrary_R2_.fastq.gz"
 - class: File
   location: "path/to/SampleTagLibrary_R1_.fastq.gz"
 - class: File
```

```
   location: "path/to/SampleTagLibrary_R2_.fastq.gz"
 - class: File
   location: "path/to/TCRBCRLibrary_R1_.fastq.gz"
 - class: File
   location: "path/to/TCRBCRLibrary_R2_.fastq.gz"
Reference_Archive:
  class: File
  location: "path/to/RhapRef_Human_WTA_2023-02.tar.gz"
Sample_Tags_Version: flex
VDJ_Version: human
```

**AbSeq-only assay:**

```
#!/usr/bin/env cwl-runner
cwl:tool: rhapsody
Reads:
 - class: File
   location: "path/to/AbSeqLibrary_R1_.fastq.gz"
 - class: File
   location: "path/to/AbSeqLibrary_R2_.fastq.gz"
AbSeq_Reference:
 - class: File
   location: "path/to/AbSeq_reference.fasta"
Putative_Cell_Call: AbSeq
```

**ATAC-Seq only assay:**

```
#!/usr/bin/env cwl-runner
cwl:tool: rhapsody
Reads_ATAC:
 - class: File
   location: "path/to/ATACLibrary_S2_L001_R1_001.fastq.gz"
 - class: File
   location: "path/to/ATACLibrary_S2_L001_R2_001.fastq.gz"
 - class: File
   location: "path/to/ATACLibrary_S2_L001_I2_001.fastq.gz"
Reference_Archive:
  class: File
  location: "path/to/RhapRef_Human_WTA-ATAC_2023-08.tar.gz"
```

#### Multiomic WTA + ATAC-Seq assay

```
#!/usr/bin/env cwl-runner
cwl:tool: rhapsody
Reads_ATAC:
 - class: File
   location: "path/to/ATACLibrary_S2_L001_R1_001.fastq.gz"
 - class: File
   location: "path/to/ATACLibrary_S2_L001_R2_001.fastq.gz"
 - class: File
   location: "path/to/ATACLibrary_S2_L001_I2_001.fastq.gz"
Reads:
 - class: File
   location: "path/to/WTALibrary_R1_.fastq.gz"
 - class: File
   location: "path/to/WTALibrary_R2_.fastq.gz"
Reference_Archive:
  class: File
  location: "path/to/RhapRef_Human_WTA-ATAC_2023-08.tar.gz"
```

## Run the Pipeline with CWL-runner

### Procedure

Local installation is supported by most Unix-like operating systems such as macOS or Linux. Minimum system requirements must be met. See System and Software Installation (page 20).

To run the pipeline on macOS, perform these additional configuration steps:

1. To enable CWL-runner to set up volumes, run the command: `$ export TMPDIR=/tmp/docker_tmp`
2. To increase the memory available to docker:
   - Click the Docker icon in the menu bar to open the docker menu.
   - Click **Preferences**, and navigate to the Advanced tab.
   - Use the slider to increase the memory limit. We recommend ≥32 GB for Targeted and ≥96 GB for WTA and ATAC-Seq. Lower limits are sufficient for smaller datasets.
   - Click **Apply & Restart** at the bottom of the window.

### Running CWL-runner

1. In the terminal, ensure that you are in a directory that contains the CWL files that were downloaded from the Bitbucket repository. The edited YML file for input specifications must also be present in this directory.

2. Run the pipeline by entering the command:

   ```
   $ cwl-runner workflow.cwl input.yml
   ```

If running the sequencing analysis pipeline, the workflow is the file `rhapsody_pipeline_<version>.cwl`, and the input specification file is the `pipeline_inputs_template_<version>.yml`.

3. If desired, you can specify the output directory for the analysis using the flag `--outdir`. An example command:

```
$ cwl-runner --outdir /path/to/results_folder rhapsody_pipeline_2.0.cwl my_sample.yml
```

**Note:** The output directory must be an existing directory. If no output directory is specified, files are output to the working directory.

4. Jobs in some steps can run in parallel. To enable this, use the flag `--parallel`. An example command:

```
$ cwl-runner --parallel --outdir /path/to/results_folder rhapsody_pipeline_2.0.cwl my_
sample.yml
```

5. Confirm that the following message displays after the pipeline is completed: `Final process status is success.`

6. Access the output files. All output files are found in the output directory specified in the CWL-runner command. If no output directory is specified, the files are output to the directory from which the command was called.

# 3

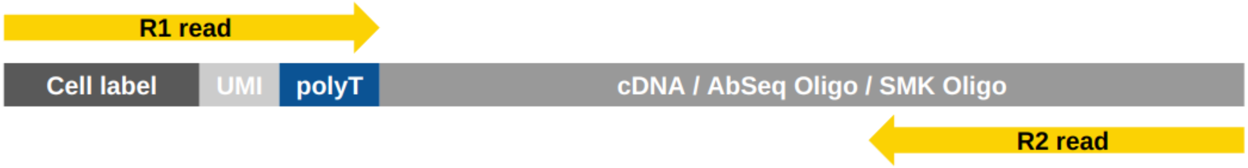
# Pipeline Steps and Algorithms

## Introduction

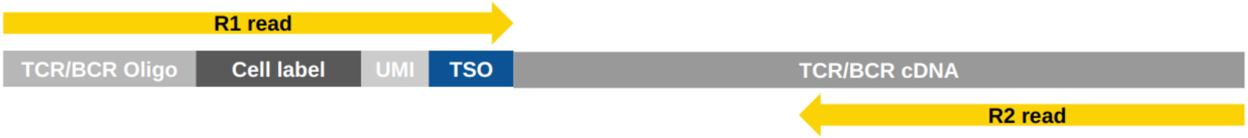
This section provides an in-depth description of each step in the BD Rhapsody™ Sequence Analysis Pipeline. The main portion of these steps decribe the WTA mRNA, Targeted mRNA, AbSeq, and Sample Tag processing steps. For details on the steps of the TCR and BCR Analysis (page 57) or ATAC-Seq Analysis (page 61), see their dedicated pages.

The BD Rhapsody™ assays are used to create sequencing libraries from single-cell multiomic experiments. For the WTA mRNA, Targeted mRNA, AbSeq, Sample Tag, TCR and BCR libraries, the analysis pipeline works with paired-end FASTQ R1 and R2 files. R1 reads contain information on the cell label and molecular identifier, and R2 reads contain information on the bioproduct. For ATAC-seq libraries the cell label information is on the I2 read, therefore the pipeline additionally requires the I2 FASTQ. The R1 and R2 reads contain information on the genomic DNA fragment. Refer to the following figures:

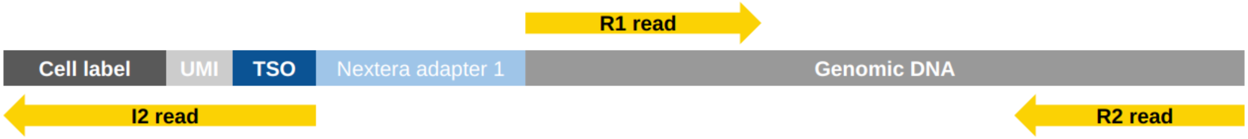
*Structure of reads generated by sequencing libraries prepared using either the BD Rhapsody™ WTA mRNA, Targeted mRNA, AbSeq or Sample Tag Assay*



*Structure of reads generated by sequencing libraries prepared using the BD Rhapsody™ TCR/BCR Full Length Assay*

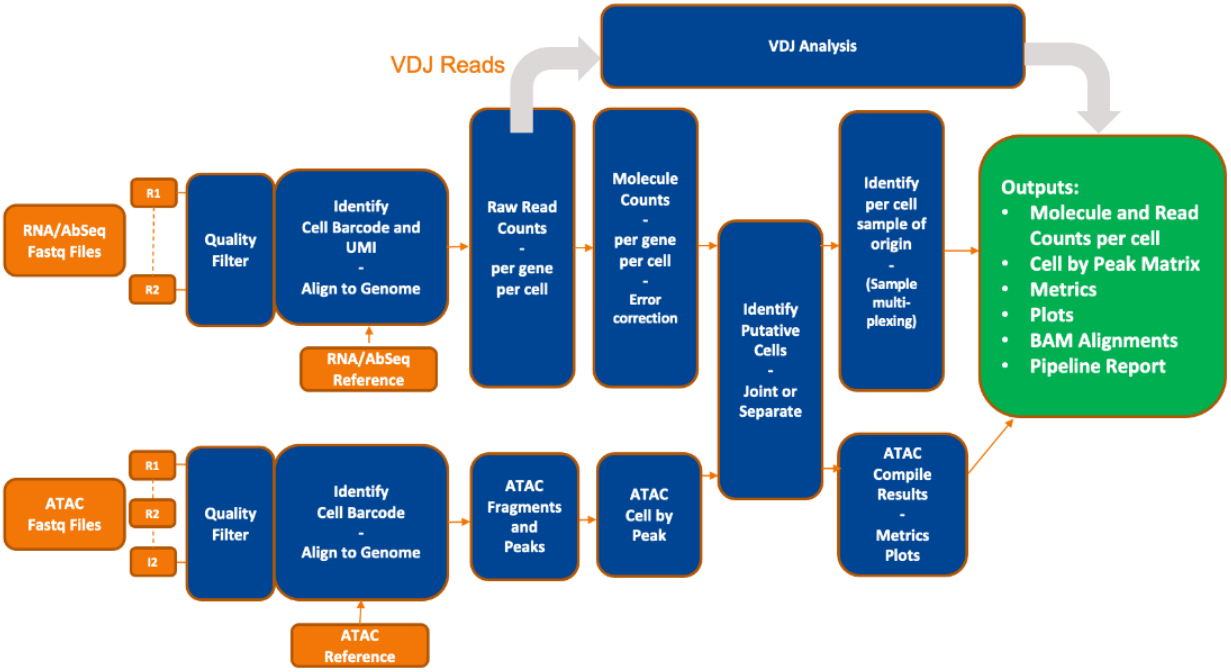


*Structure of reads generated by sequencing libraries prepared using the BD Rhapsody™ ATAC-Seq Assay*



## Pipeline overview

After sequencing, the pipeline takes input from FASTQ files, a reference (Targeted panel or WTA / WTA+ATAC-Seq reference archive), an AbSeq reference (if required), and a supplemental reference (if required), using these to generate output files and metrics about the pipeline run.


VDJ Reads
VDJ Analysis
RNA/AbSeq Fastq Files
R1
R2
Quality Filter
Identify Cell Barcode and UMI - Align to Genome
RNA/AbSeq Reference
Raw Read Counts - per gene per cell
Molecule Counts - per gene per cell - Error correction
Identify Putative Cells - Joint or Separate
Identify per cell sample of origin - (Sample multi-plexing)
Outputs:
• Molecule and Read Counts per cell
• Cell by Peak Matrix
• Metrics
• Plots
• BAM Alignments
• Pipeline Report
ATAC Fastq Files
R1
R2
I2
Quality Filter
Identify Cell Barcode - Align to Genome
ATAC Reference
ATAC Fragments and Peaks
ATAC Cell by Peak
ATAC Compile Results - Metrics Plots

The steps described in this section comprise the following:

1. Read Quality Filter (page 29)
2. Annotate R1 Cell Label and UMI (page 31)
3. Align and Annotate R2 Reads (page 33)
4. Annotate molecules and remove artifacts—see Molecules and Error Correction (page 35)
5. Determine Putative Cells (page 41)
6. Determine the sample of origin (sample multiplexing only). See Sample Tag Analysis (page 52)
7. Generate Expression Matrix Data Tables (page 55)
8. Annotate BAM (page 56)
9. Perform TCR and BCR Analysis (page 57) if applicable
10. Perform ATAC-Seq Analysis (page 61) if applicable

# Read Quality Filter

## Read overlap detection

First, read 1 and read 2 are tested to see if they overlap, so that read 1 content can be removed from read 2. This will prevent downstream mis-alignment and mis-assembly of any cell label sequences present in read 2. An overlap detection percent metric is calculated and may help troubleshoot PCR cleanup and library preparation steps. This overlap step does not remove any read pairs from subsequent steps.

Read 1 artifacts are removed from read 2 with the following steps:

- Read 1 and 2 are compared with a modified Knuth-Morris-Pratt substring search algorithm that allows for a variable number of mismatches. The maximum mismatch rate is set to 9% by default with a minimum overlap length of 25 bases. Read 1 is scanned right to left on the reverse complement of read 2. The closest offset from the end of the reverse complement of read 2 with the lowest number of mismatches (below the maximum mismatch rate threshold) is considered to be the best fit overlap.
- The merged read will be split back into a read pair. The merged read will be split according to the bead specific R1 minimum length (described in Annotate R1 Cell Label and UMI (page 31)). The bases at the beginning of the merged read up to the R1 minimum length, plus the length of the bead capture sequence, will be assigned to read 1, and the rest will be assigned to read 2.

## Read trimming

Then, read 1 and read 2 are trimmed in these ways:

Read 1:

1. Remove sequence longer than necessary for cell label and UMI identification (length kept depends on bead version)
2. Remove bead sequence - 5' TCR/BCR primer: `ACAGGAAACTCATGGTGCGT`

Read 2:

1. Remove 3' poor quality bases if quality scores go below 20, using the BWA Quality Trimming Algorithm
2. Remove bead sequence - 5' TCR/BCR primer: `ACAGGAAACTCATGGTGCGT` and template switch oligo (TSO): `TATGCGTAGTAGGTA` or `GTGGAGTCGTGATTATA`

## Filtering criteria

Finally, the following filtering criteria are applied to each read pair:

| Bead | **Minimum Read 1 Length** | **Minimum Read 2 Length** | **Minimum Mean Base Quality** | **R1 Single Nucleotide Frequency** | **R2 Single Nucleotide Frequency** |
|---|---|---|---|---|---|
| Original V1 | 60 | 40 | 20 | 0.55 | 0.8 |
| Enhanced 3' | 43 | 40 | 20 | 0.55 | 0.8 |

| Bead | **Minimum Read 1 Length** | **Minimum Read 2 Length** | **Minimum Mean Base Quality** | **R1 Single Nucleotide Frequency** | **R2 Single Nucleotide Frequency** |
|---|---|---|---|---|---|
| Enhanced TCR/BCR | 63 | 40 | 20 | 0.55 | 0.8 |

- **Read length**: If the length of the R1 read is less than the bead-specific R1 minimum length (described in **Annotate cell label and UMI**) or the R2 read is <40 bp, the R1/R2 read pair is dropped.
- **Mean base quality score of the read**: If the mean base quality score of either the R1 read or the R2 read is <20, the read pair is dropped.
- **Highest Single Nucleotide Frequency (SNF) observed across the bases of the read**: If the SNF is ≥0.55 for the R1 read or the SNF is ≥0.80 for the R2 read, the read pair is dropped. This criterion removes reads with low complexity such as strings of identical bases and tandem repeats.

The thresholds for each filter are determined empirically.

Reads are tested against each filter in the following order: Read length, Single nucleotide frequency, and Mean base quality. Reads that fail one filter are removed and not tested in subsequent filters.

# Annotate R1 Cell Label and UMI

## R1 structure

The quality-filtered R1 reads are analyzed to identify the cell label sequences (CLS), common linker sequences (L), and Unique Molecular Identifier (UMI) sequence.

Read 1 structure by bead version:

| Bead Version | TCR/ BCR Handle | Diversity Insert | CLS1 | Linker1 | CLS2 | Linker2 | CLS3 | UMI | Capture Sequence |
|---|---|---|---|---|---|---|---|---|---|
| Original V1 | None | None | 9bp | ACTGGC<br>CTGCGA | 9bp | GGTAGC<br>GGTGACA | 9bp | NNNNNNNN | 18 dT |
| Enhanced 3' | None | None, A, GT, or TCA | 9bp | GTGA | 9bp | GACA | 9bp | NNNNNNNN | 25 dT |
| Enhanced TCR/BCR | ACAGG<br>AAACT<br>CATGG<br>TGCGT | None | 9bp | AATG | 9bp | CCAC | 9bp | NNNNNNNN | TATGCG<br>TAGTAG<br>GTATG<br>or<br>GTGGAG<br>TCGTG<br>ATTATA |

## Cell label

Information of the cell label is captured by bases in three sections (CLS1, CLS2, CLS3) along each R1 read. Two common sequences (L1, L2) separate the three CLSs, and the presence of L1 and L2 relates to the way the capture oligo nucleotide probes on the beads are constructed. By design, each CLS has one of either 96 or 384 predefined sequences (depending on bead version), which has a Hamming distance of at least four bases and an edit distance of at least two bases apart. A cell label is defined by the unique combination of predefined sequences in the three CLSs. Thus, the maximum possible number of cell labels is either $96^3$ or $384^3$. In the final data tables, the three part cell label is converted to a single integer index between 1-$384^3$.

Reads are first checked for perfect matches in all three pre-designed CLS sequences at the expected locations, and reads with perfect matches are kept.

The remaining reads are subjected to another round of filtering to recover reads with base substitutions, insertions, and deletions caused by sequencing errors, PCR errors, or errors in oligonucleotide synthesis.

## UMI

By design, the UMI is a string of eight randomers immediately downstream of CLS3. For reads with insertions or deletions within the CLSs, the UMI sequence is eight bases immediately following the end of the identified CLS3.

## Cell label sequences and utility functions

Cell label structure, cell label sequences, bead sequences, and python utility functions are available for download here:

rhapsody_cell_label.py.txt

The single integer cell index and cooresponding assembled R1 bead sequence is available in fasta format for each bead version:

- Original V1 beads: Rhapsody_cellBarcodeV1_IndexToSequence.fasta.zip
- Enhanced beads: Rhapsody_cellBarcodeEnh_IndexToSequence.fasta.zip
- Enhanced V2 beads: Rhapsody_cellBarcodeEnhV2_IndexToSequence.fasta.zip

# Align and Annotate R2 Reads

## Alignment to reference

Read pairs that pass quality filters and have a valid cell label and UMI on R1 have their R2 aligned to a reference using STAR (Spliced Transcripts Alignment to a Reference © Alexander Dobin, 2009-2022). For Targeted assays, a STAR reference is generated dynamically based on the input Targeted_Reference FASTA file, and any other provided FASTA files. For WTA assays, the STAR reference is prebuilt and provided in the Reference_Archive input, and any additional FASTA files are included as genome FastaFiles for alignment.

## Criteria for a valid R2 read

Targeted assays:

For targeted assays, an R2 read is a valid alignment if all of these criteria are met:

- The R2 alignment begins within the first five nucleotides for mRNA, first 15 nucleotides for AbSeq, and first 25 nucleotides for Sample Tags. This criterion ensures that the R2 read originates from an actual PCR priming event.
- The length of the alignment match (can be a match or mismatch) in the CIGAR string is >=37 for mRNA >=25 for AbSeq and >=40 for Sample Tags. A CIGAR (Compact Idiosyncratic Gapped Alignment Report) string is a sequence of base lengths to indicate base alignments, insertions, and deletions with respect to the reference sequence.
- The read does not align to phiX174.

WTA assays:

By default, alignments to both exons and introns are used. Including reads that align to introns may increase sensitivity, resulting in an increase in molecule counts and the number of genes per cell for both cellular and nuclei samples. Reads that align to introns may indicate the presence of unspliced mRNAs and are also useful in the study of nuclei and RNA velocity.

An R2 is a valid gene alignment if all of these criteria are met:

- The sum of the CIGAR alignment matches must be >=25.
- The read aligns uniquely to an exon or intron of a bioproduct in the reference.
- The read does not align to phiX174.
- If "Exclude Intronic Reads" option is selected, read must align to exon.

# Valid Read Pairs

Read pairs with a valid R1 read and a valid R2 read are retained for further analysis.

A valid R1 read requires identified CLSs, and a UMI sequence with non-N bases.

A valid R2 read must uniquely map to the exon or intron of a bioproduct in the reference. For targeted, it must also have the correct PCR2 primer sequence at the start and an alignment match sufficient in length.

# Molecules and Error Correction

## Collapse reads into raw molecules

Reads with the same cell label, same UMI sequence, and same bioproduct are collapsed into a single raw molecule. The number of reads associated with each raw molecule is reported as the ***raw adjusted sequencing depth***.

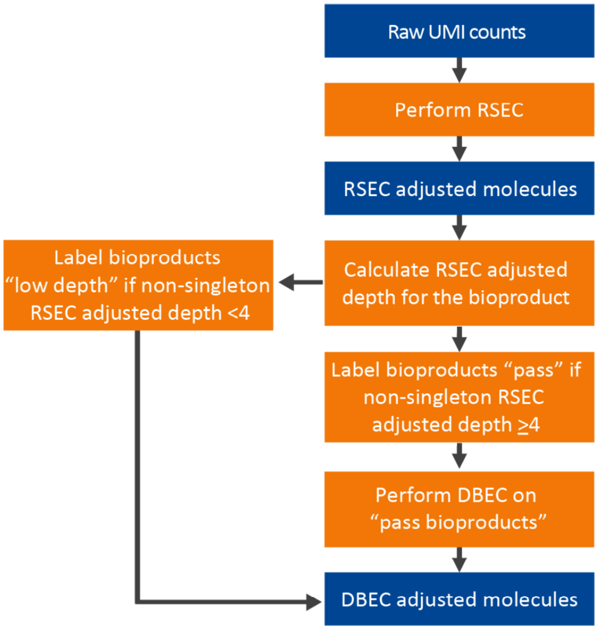
## Remove artifact molecules using RSEC and DBEC UMI adjustment algorithms

PCR and sequencing often generate errors. If the error occurs within the UMI sequence, the R1/R2 read pair is called a unique molecule but is, in fact, an artifact. Artifact molecules contribute to an over-estimated molecule count of a gene in a cell. As sequencing depth increases, the number of raw molecules rises and never plateaus due to these artificial molecules.

To remove the effect of UMI errors on molecule counting, BD Biosciences has developed a set of UMI adjustment algorithms. UMI errors that are single base substitution errors are identified and adjusted to the parent UMI barcode using recursive substitution error correction (RSEC). For targeted sequencing analysis, other UMI errors derived from library preparation steps or sequencing base deletions are later adjusted using distribution-based error correction (DBEC).

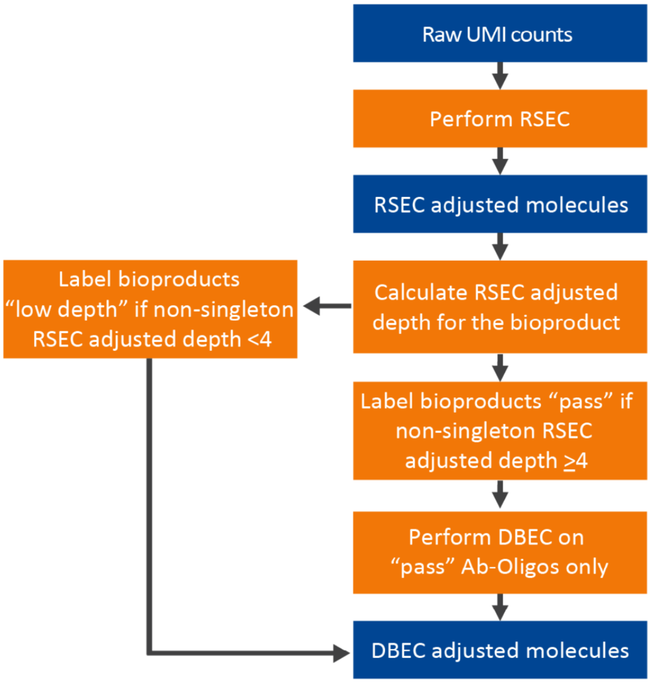
Note that targeted sequencing analysis uses RSEC and DBEC, while WTA sequencing analysis uses RSEC only for mRNA libraries and RSEC and DBEC for AbSeq libraries.

**Targeted assay workflow** of UMI count adjustment -- using RSEC and DBEC algorithms on both mRNA and AbSeq:


Raw UMI counts
Perform RSEC
RSEC adjusted molecules
Calculate RSEC adjusted depth for the bioproduct
Label bioproducts "low depth" if non-singleton RSEC adjusted depth <4
Label bioproducts "pass" if non-singleton RSEC adjusted depth ≥4
Perform DBEC on "pass bioproducts"
DBEC adjusted molecules

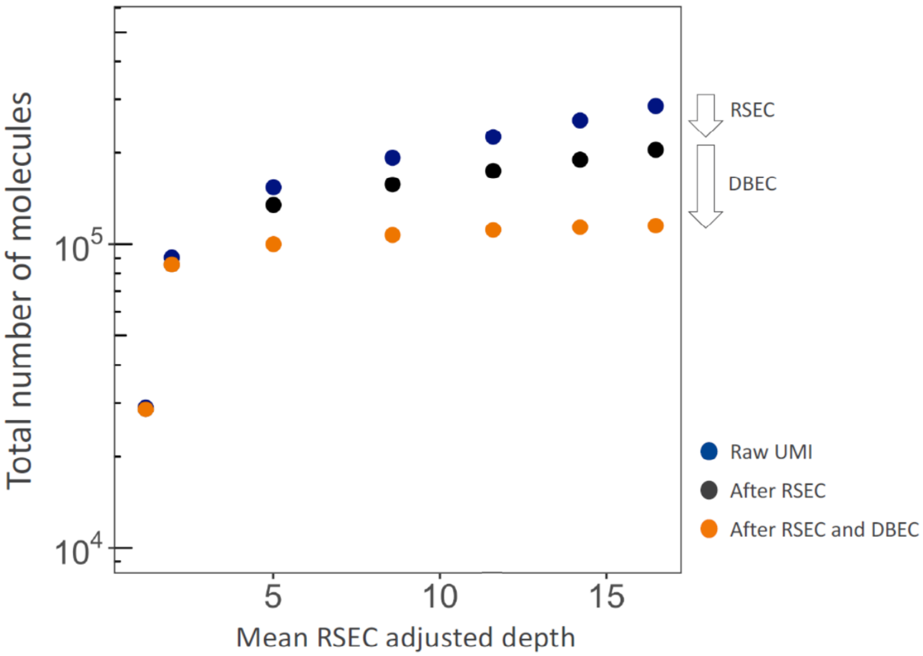
Targeted workflow

**WTA assay workflow** of UMI count adjustment -- using RSEC on mRNA, and RSEC and DBEC algorithms on AbSeq:


Raw UMI counts
Perform RSEC
RSEC adjusted molecules
Calculate RSEC adjusted depth for the bioproduct
Label bioproducts "low depth" if non-singleton RSEC adjusted depth <4
Label bioproducts "pass" if non-singleton RSEC adjusted depth ≥4
Perform DBEC on "pass" Ab-Oligos only
DBEC adjusted molecules

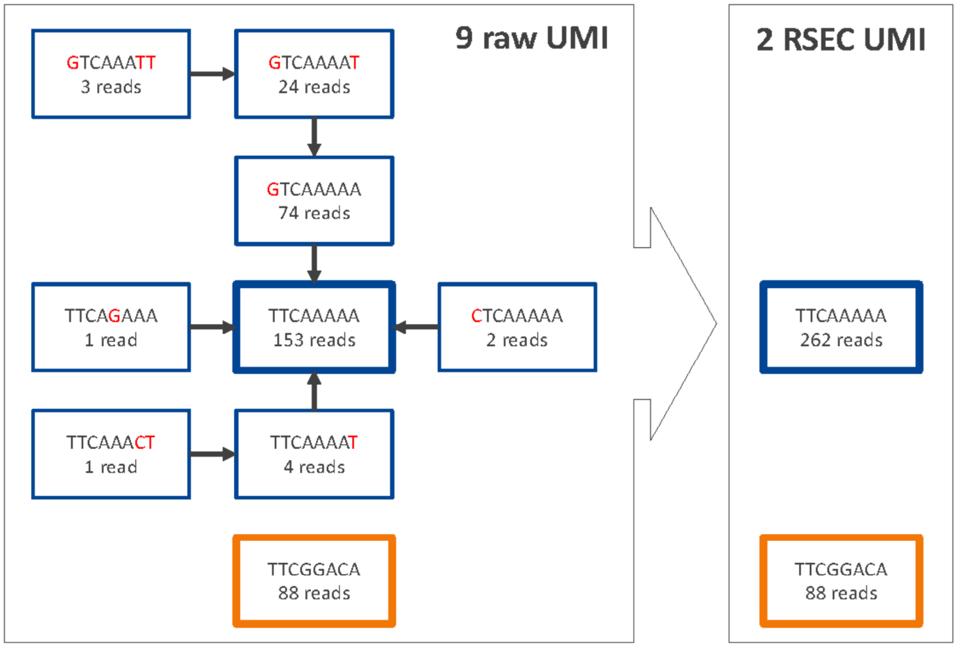
WTA workflow

The following graph shows the impact on total molecule count of applying RSEC and DBEC to an example dataset. For targeted sequencing analysis, if we consider only raw UMIs, the apparent total number of molecules continues to rise with sequencing depth, because the presence of sequencing and PCR errors contribute to unique UMIs. RSEC removes artifact molecules from single base substitutions in the UMI sequence. Further adjustment by DBEC removes artifact molecules originating from PCR errors. As a result, the number of molecules stabilizes with additional sequencing, indicating the library is sequenced to saturation.



## RSEC algorithm to collapse molecules that differ by one base in the UMI sequence

RSEC considers two factors in error correction: 1) similarity in UMI sequence and 2) raw UMI coverage or depth. The following is a somewhat extended example of the recursive error correction algorithm in practice, wherein nine raw UMIs are collapsed into two corrected UMIs.



For the molecules from each combination of cell label and bioproduct, UMIs are connected when their UMI sequences are matched to within one base (Hamming distance = 1). For each connection between UMI x and y, if Coverage(y)>2 *Coverage(x)– 1, then y is the Parent UMI and x is the Child UMI. Based on this assignment, child UMIs are collapsed to their parent UMI. This process is recursive until there are no more identifiable parent-child UMIs for the bioproduct.

The number of reads for each child UMI is added to the parent, so no reads are lost. The sum of the reads is the ***RSEC-adjusted depth*** of the ***RSEC-adjusted molecule***.

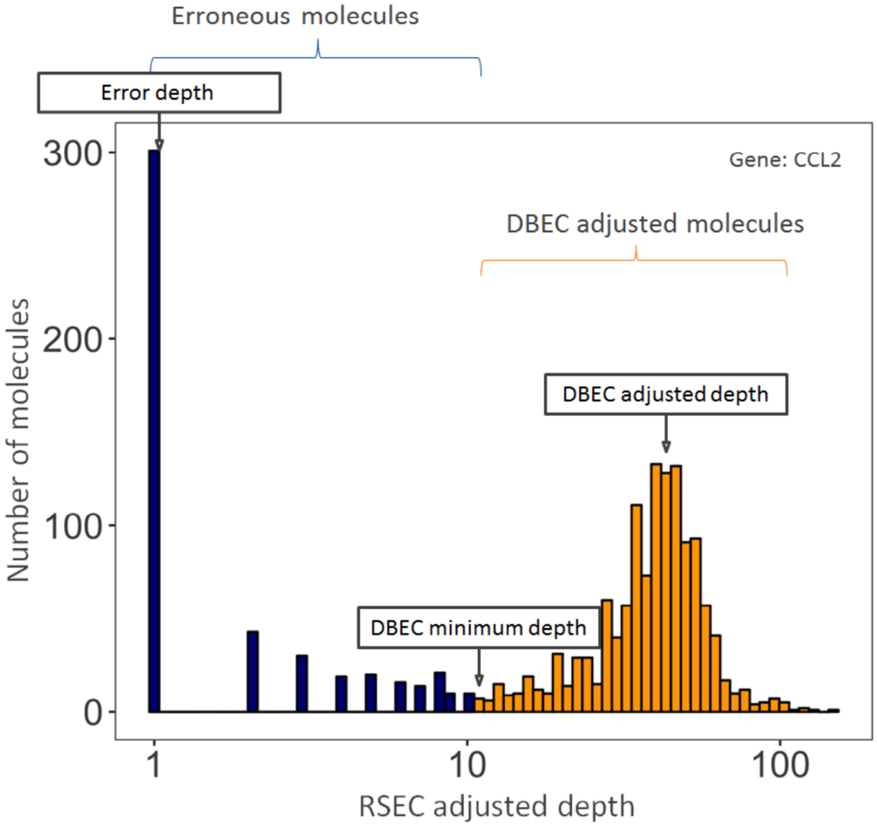
## DBEC algorithm to further adjust molecule counts by bioproduct

The RSEC-adjusted molecule counts are further corrected by DBEC, depending on assay type. For Targeted assays, DBEC is applied on all bioproduct types (mRNA and AbSeq). For WTA assays, DBEC is applied only to AbSeq targets.

DBEC is applied on a per-bioproduct basis. The algorithm is based on the assumption that the targeted PCR amplified set of molecules of the same bioproduct, regardless of the cell of origin, is subject to the same amplification efficiency, and therefore, should have similar read depth. Artifact molecules created later in the PCR cycles, such as those derived from PCR chimera formation, will likely have less read depth.

DBEC considers the distribution of RSEC-adjusted depth distribution, not UMI sequence. The sequencing depth of RSEC-adjusted molecules for each bioproduct is a bimodal distribution. The lower mode of the distribution likely represents artifact molecules, and the upper mode likely represents true molecules. The algorithm fits two negative binomial distributions to statistically distinguish between the two modes. Molecules in the upper mode are retained (***DBEC-adjusted molecules***), while the molecules in the lower mode are discarded. The average depth of the molecules in the upper mode is known as the ***DBEC-adjusted seq depth***. The cutoff between the two modes is the ***DBEC minimum depth***.

See the following example figure for gene CCl2. Counts under the orange bars are kept and labeled as DBEC-adjusted molecules. Counts under the blue bars are labeled as erroneous molecules and are discarded. The error depth and DBEC-adjusted depth arrows point to the respective average depths.



DBEC is applied to bioproducts with an average non-singleton RSEC sequencing depth ≥4. This means that the depth is calculated after removing RSEC UMIs with only one representative read. According to the Poisson distribution, if the average UMI depth is <4, more signal UMIs are removed than error UMIs. As a result, a bioproduct is marked as *pass* if its average RSEC depth ≥4 and is subject to DBEC. Otherwise, it is marked *low depth* and bypasses DBEC. If no count is associated with the bioproduct, it is labeled as *not detected*.

DBEC removes molecules and the reads associated with the removed molecules from consideration in downstream analyses.

The RSEC and DBEC metrics associated with each bioproduct are reported in the file, `<sample_name>_Bioproduct_Stats.csv`.

# Determine Putative Cells

The BD Rhapsody™ Sequence Analysis Pipeline provides the following options to determine putative cells:

1. Basic algorithm for putative cell identification using second derivative analysis (page 41)
2. Refined algorithm for adjusting putative cell counts for mRNA and AbSeq (page 44)
3. Refined algorithm for adjusting putative cell counts for ATAC-Seq (page 45)

Ideally, the number of unique cell labels detected by the BD Rhapsody™ Sequence Analysis Pipeline should be similar to the number of cells captured and amplified by the BD Rhapsody™ workflow. However, various processes throughout the workflow can introduce noise that contributes to extra cell labels seen during sequencing analysis, including:

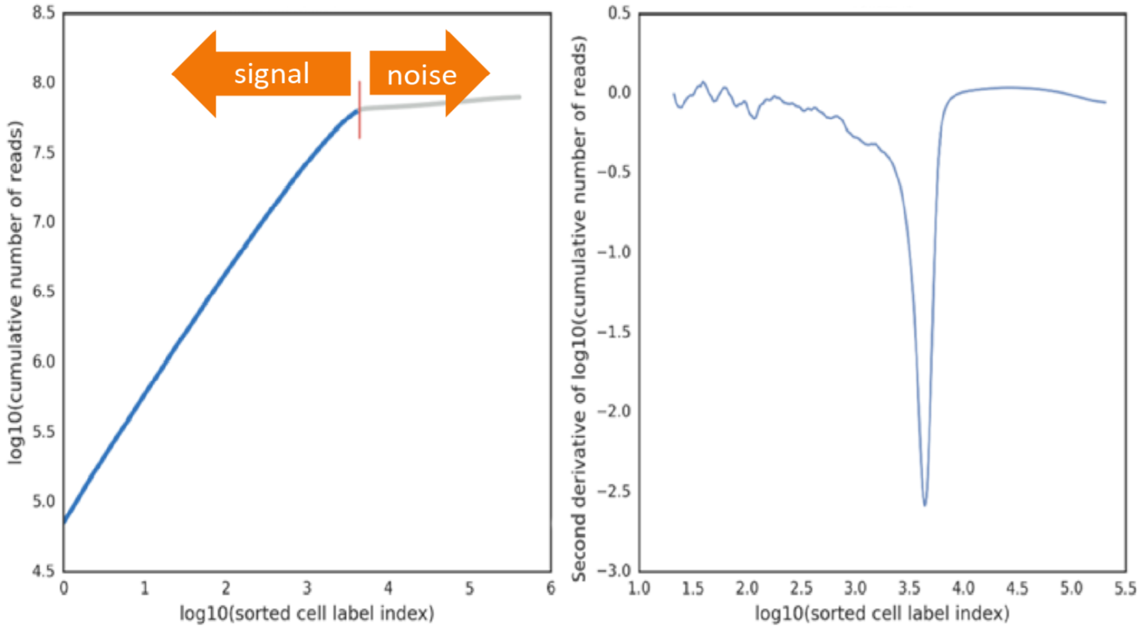
- Hybridization of polyadenylated (polyA) oligonucleotides to non-cell beads residing in neighboring wells when the cell lysis step is too long
- Under-loading beads in BD Rhapsody™ cartridges, resulting in cells in a well without a bead, and thus the RNA from the cells diffusing to adjacent wells
- Low-level oligo contamination during bead synthesis
- Errors generated during the PCR amplification steps of the workflow

To distinguish cell labels associated with true putative cells from those associated with noise, the basic cell calling algorithm is used by default for all assays. When improved sensitivity or specificity of detecting putative cells is desired, refined algorithms are available for mRNA/AbSeq and ATAC-Seq.

## Basic algorithm for putative cell identification using second derivative analysis

The principle of the basic cell calling algorithm is that cell labels from actual cell capture events should have many more reads associated with them than noise cell labels. For mRNA/Abseq data, the basic algorithm is performed using read counts from mRNA molecules (by default) or from Abseq. All reads associated with RSEC adjusted molecules from the selected bioproduct type (mRNA by default) are taken into account. For ATAC-Seq, the basic algorithm is performed using the number of transposase sites in peaks.

In the following figure, the number of mRNA/AbSeq reads of each cell is plotted on a log10-transformed cumulative curve, with cells sorted by the number of reads in descending order (left image). Additionally, the rate of change of the cumulative count is calculated with the second derivative (right image). In a typical experiment, a distinct inflection point is observed in the second derivative, indicating a division between signal cell labels and noise cell labels (red vertical line). The algorithm finds the minimum second derivative along the cumulative read curve as the inflection point. Cell labels to the left of the red vertical line are most likely derived from a cell capture event and are considered as signal. The remaining cell labels to the right of the red line are noise. This is the basic algorithm result and is the default cell calling algorithm.



If every cell in the sample is well represented by molecules from library preparation, there is only one inflection point. The number of reads of the putative cells is a single distribution well separated from the noise distribution.

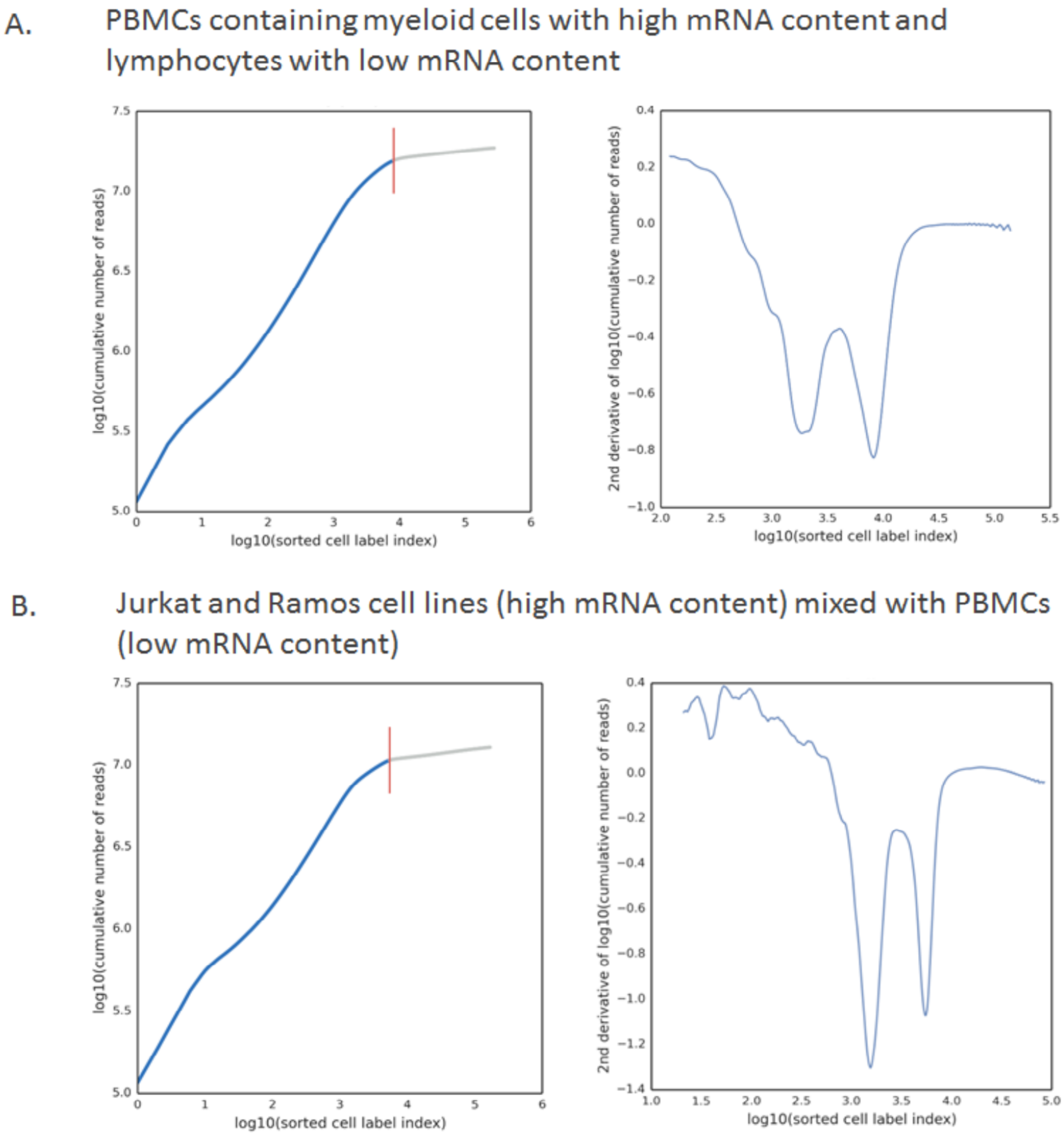
There are situations, however, when a sample contains cells with a very wide range of number of molecules. If sub-populations of cells with high and low mRNA content are considerably large, multiple inflection points can be observed. Example scenarios include biological samples such as peripheral blood mononuclear cells (PBMCs) with myeloid cells being much larger and active carrying thousands of molecules compared to lymphocytes being smaller and less active carrying tens of molecules (A. in the following figure), or artificial mixtures of cell line cells and primary cells (B. in the following figure). Targeted panels with an uneven representation of genes expressed in different cell types of a sample can also lead to multiple inflection points.

Inflection points are considered valid if:

- The second derivative minimum corresponding to the inflection point is at least half as deep as the global minimum and is ≤–0.3
- The inflection point is within a dynamically determined range of number of putative cells. The lower threshold is fixed at 25 cells. The upper threshold on the number of putative cells is:

- ○ For V1 and Enhanced beads: 25% of the number of total cell labels with read counts >=10
- ○ For Enhanced V2/V3 beads: 40% of the number of total cell labels with read counts >= 10
- ○ The upper threshold will not go below 50,000 cells

The smoothing window of the second derivative curve increases until there are two valid inflection points. By default, the valid inflection point corresponding to the larger cell number is deemed the better one. When the `Expected_Cell_Count` pipeline parameter is set, the selected inflection point is whichever one is nearest to the expected cell count value. Usually this can be set to the number of cells loaded into the BD Rhapsody™ cartridge.



A. PBMCs containing myeloid cells with high mRNA content and lymphocytes with low mRNA content

B. Jurkat and Ramos cell lines (high mRNA content) mixed with PBMCs (low mRNA content)

## Refined algorithm for adjusting putative cell counts for mRNA and AbSeq

In some cases, the basic implementation of the second derivative analysis might include small numbers of false positive and false negative cell labels. If the refined algorithm is selected, additional refinement steps are run to attempt to identify these false positive and false negative cell labels.

### Removing false positives

Consider the case where the chosen inflection point includes the populations of cell labels with wide ranges of number of reads per cell label. Then, the signal population with lower reads per cell label might also include noise cell labels derived from residual mRNA molecules from the cells with very high mRNA content. The number of reads associated with these noise cell labels derived from high-expressing cells can be indistinguishable from low-expressing cells, which have similar reads per cell.

Since these false positive cells can be hard to identify with reads alone, the relative expression profile of cell labels can be used to identify them. For example, a false positive cell label that is derived from a high mRNA-expressing, true positive cell label would likely have a similar expression profile but with a lower read signal. Therefore, a second derivative analysis is done on the most variable genes to identify these false positive cell labels.

The most variable gene expression is defined by a process similar to that described by Macosko, EZ, *et al.* (see References) :

1. Log-transform read counts of each gene within each cell to get the gene expression: log10 (count + 1).

2. Calculate the mean expression and dispersion (defined as variance/mean) for each gene.

3. Place genes into 20 bins based on their average expression.

4. Within each bin, calculate the mean and standard deviation of the dispersion measure of all genes, and then calculate the normalized dispersion measure of each gene using the following equation:

   ```
   Normalized dispersion = (dispersion – mean)/(standard deviation)
   ```

5. Apply a cutoff value for the normalized dispersion to identify genes for which expression values are highly variable even when compared to genes with similar average expression.

A second derivative analysis is applied on variable gene sets defined by a different cutoff value for the normalized dispersion to derive the cell label filtered set B. For each dispersion cutoff, the noise cell labels are determined as A–B. For instance, for three cutoff values, noise cell labels are N1 = A–B1, N2 = A–B2, and N3 = A–B3, where the minus sign represents the set difference. The common noise cell labels detected among N1, N2, and N3 are subtracted from cell labels set A (those identified by the basic algorithm). The resultant set is denoted as cell label filtered set C = A – intersection(N1, N2, N3).

### Recovering false negatives

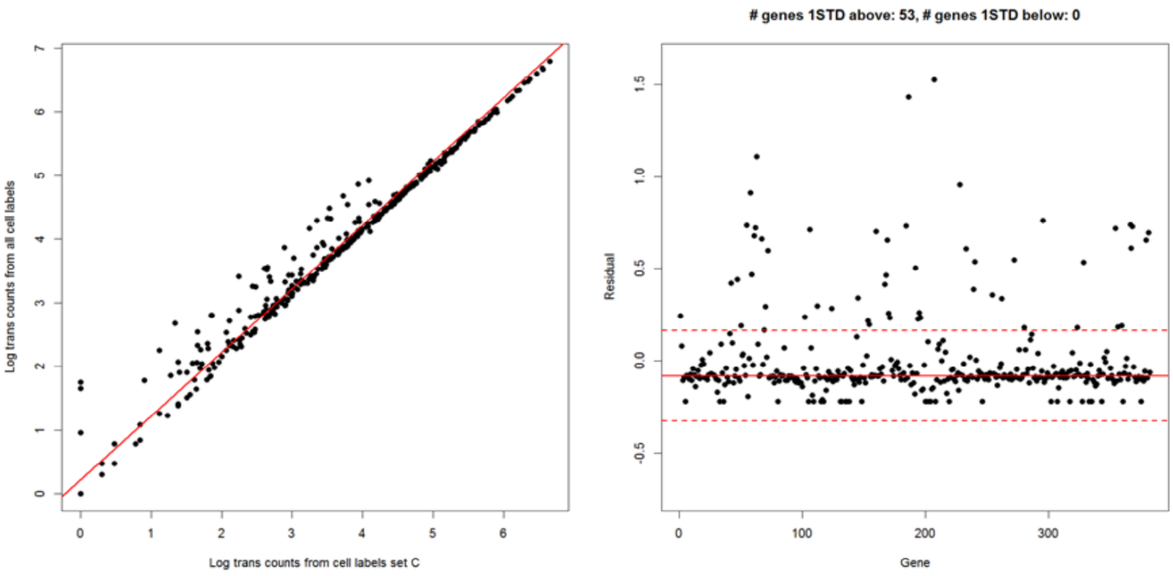
Cells with low numbers of molecules might be missed by the basic implementation of the second derivative analysis algorithm, because a cell subset might express very few of the genes in the (Targeted assay) gene list. The cell labels carry a very low number of reads, and the size of the cell population is small enough that their cell labels do not form a distinct second inflection point. These cell labels might be mistaken as noise.

If there are genes specific to the false negative cell label subset (for example, marker genes), they can be identified by comparing the number of reads for each gene from all detected cell labels to those from cell labels deemed as signal. The assumption is that the relative abundance of reads for each gene from all of the noise cell labels should be no different than that from all of the cell labels considered as signal. If a specific cell subset is missed initially, there is a set of genes that appears as enriched in the noise cell labels in the basic implementation.

Left and right images in the following figure: Detecting genes enriched in noise as determined by the basic implementation of the second derivative analysis. Each dot represents a gene. This enriched set of genes is detected by the following steps:

1. For each gene, calculate the total read counts from all detected cell labels and from cell labels in set C.
2. Identify the genes that have the biggest discrepancy in representation by cell labels in set C versus all cell labels. This is done by plotting and finding the line of best fit to detect the genes with the largest residuals at least one standard deviation away from the median of residuals of all genes.

The two red dashed lines correspond to one standard deviation above and below the median (red solid line). In this example, 53 genes are enriched in the noise population.



The second derivative analysis algorithm is run again with this enriched set of genes. The recovered cell labels (*cell label filtered set D*) are combined with cell labels in set C to form set E. As a final cleanup step, cell labels carrying less than the dynamically determined minimum threshold number of molecules are removed. The number of cell labels in the final set is the number of putative cells from the refined algorithm.

## Refined algorithm for adjusting putative cell counts for ATAC-Seq

In some cases, the basic implementation of the second derivative analysis might include small numbers of false positive and false negative cell labels. If the data quality is not ideal, it may be difficult to separate the signal from the noise using only the number of transposase sites in peaks. If the ATAC-Seq refined algorithm is selected, a Guassian Mixture Model (GMM) approach is used to determine putative cells using two metrics: the number of transposase sites in peaks and the fraction of transposase sites in peaks.

The ATAC-Seq refined algorithm has five main steps:

1. Filter out low-quality cell labels using the fraction of transposase sites in peaks (page 46)
2. Filter out low-quality cell labels using the number of transposase sites in peaks (page 46)
3. Identify a putative cell cluster using the number of transposase sites in peaks - the first GMM (page 48)
4. Identify a putative cell cluster using the number of transposase sites in peaks and fraction of transposase sites in peaks - the second GMM (page 48)
5. Refine the boundary of the putative cell cluster (page 49)

The first two steps filter out low-quality cells, the third step identifies a putative cell cluster using the number of transposase sites in peaks (the first GMM), the fourth step identifies a putative cell cluster using both the number of transposase sites in peaks and the fraction of transposase sites in peaks (the second GMM), and the last step refines the boundary of the putative cell cluster. The first two steps filter out low-quality cell labels using the fraction of transposase sites in peaks and the number of transposase sites in peaks, respectively. The filtering in steps 1 and 2 improve the fit of the first and second GMMs in steps 3 and 4 where the boundary of the putative cell cluster is determined. The last step further refines the boundary of the putative cell cluster, either by recovering false negative cell labels or by filtering out false positive cell labels.

## Filter out low-quality cell labels using the fraction of transposase sites in peaks

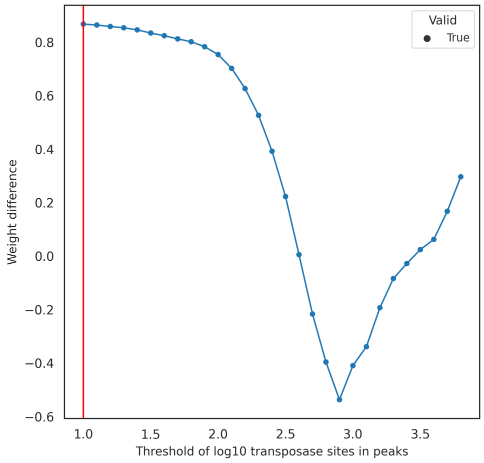
The cell labels with a low fraction of transposase sites in peaks show low specificity in targeting open chromatin sites. Those cell labels might have been generated due to several reasons such as inefficient transposase activity or non-ideal cell conditions. The threshold for the fraction of transposase sites in peaks is 0.1. Any cell labels with a fraction of transposase sites in peaks below 0.1 are filtered out.

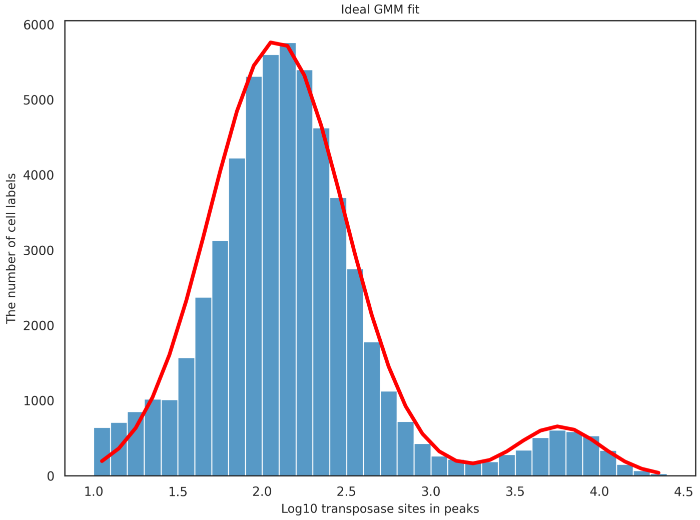
## Filter out low-quality cell labels using the number of transposase sites in peaks

The cell labels with a low number of transposase sites in peaks show low sensitivity in the assay. The threshold for the minimum number of transposase sites in peaks is dynamically selected for each dataset as follows. The bottom and top 1% of the log10-transformed number of transposase sites in peaks are calculated and used as the lower and upper bounds of the dynamic threshold testing range. If the lower bound value is less than 1, meaning less than 10 transposase sites in peaks, the lower bound is set to 1. This adjustment is to avoid selecting an extremely small value for the threshold and to ensure that cell labels with low sensitivity are filtered out.

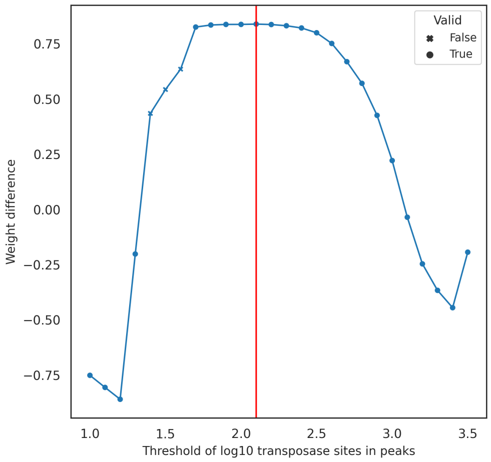
Each possible threshold from the lower bound to the upper bound at a 0.1 interval is tested using the GMM approach. For each tested threshold, the cell labels having transposase sites in peaks below the threshold are filtered out, and the remaining cell labels are fitted into two clusters using the GMM approach. The difference of the GMM weights of the two clusters is calculated by subtracting the weight of the putative cell cluster (the cluster with a higher 90% confidence interval) from the weight of the non-cell cluster. If the 90% confidence intervals of the two clusters completely overlap, the tested threshold is considered invalid and is not selected for the final number of transposase sites in peaks threshold. For good-quality data (left plot in the following figure), the weight difference usually decreases as the tested threshold value increases, because more cell labels from the non-cell cluster are filtered out. Then, the weight difference starts to increase when the majority of non-cell labels are filtered out and therefore the GMM starts to detect two sub-clusters within the putative cell cluster.

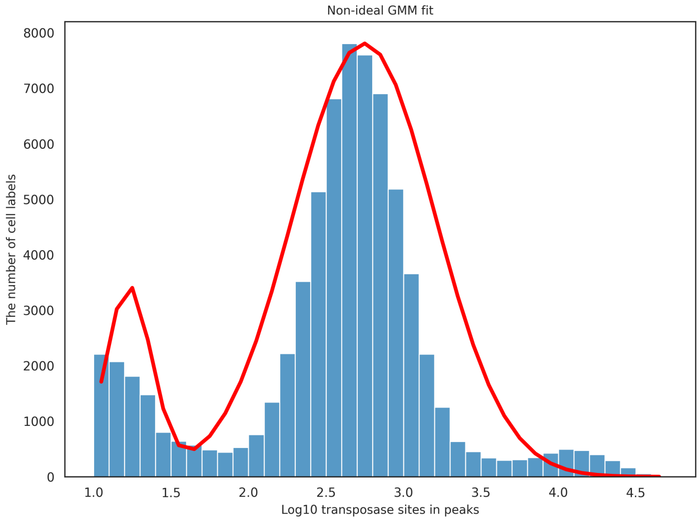
The following figures show an ideal GMM fit (right figure) when low-quality cells are properly filtered out using the final number of transposase sites in peaks threshold determined (left figure, red vertical line).





However, low-quality data with lots of noisy cell labels shows a different trend: as the tested threshold value of the number of transposase sites in peaks increases, the weight difference initially increases and then later follows the trend of good-quality data (left plot in the following figure). That initial increase is due to the abundance of noisy cell labels which causes the GMM to miss the putative cell cluster and instead identify two clusters in the non-cell labels, as shown in the right plot. However, after the majority of non-cell labels are filtered out, the GMM starts to detect a putative cell cluster, causing the weight difference to plateau and then decrease. When the majority of non-cell labels are filtered out, the weight difference increases, similar to the good-quality data shown in the preceding plots.





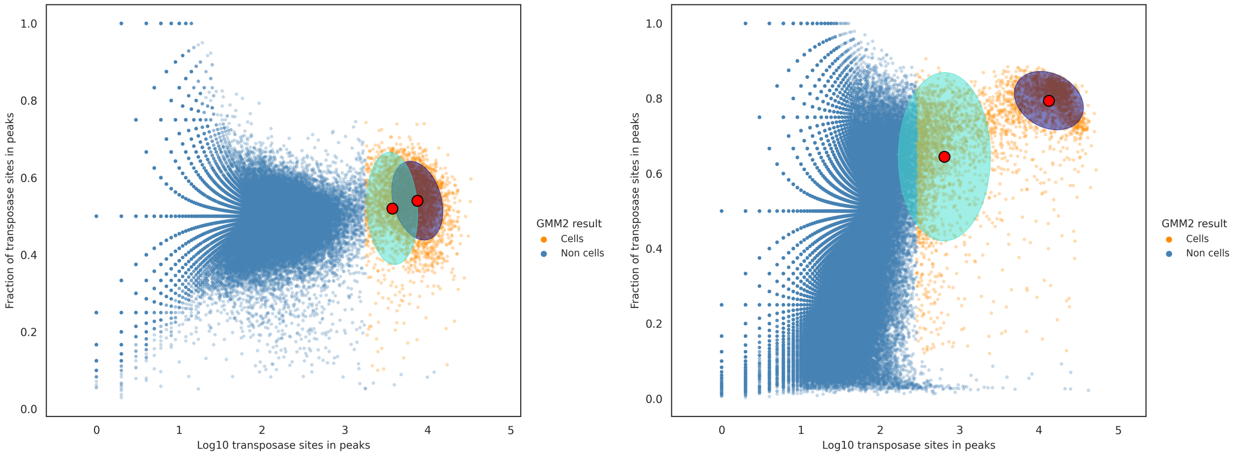
Among the valid thresholds, the threshold with the largest GMM weight difference is selected for the threshold of the number of transposase sites in peaks (red vertical line in the preceding images) and used to filter out the low-quality cell labels. The rationale is that the weight difference is a rough estimate that shows when both a non-cell cluster and a putative cell cluster are detectable without losing many non-cell labels. If the weight difference is small, many non-cell labels may be filtered out. This may lead to the undesirable result where the GMM fits both clusters to the putative cell cluster, resulting in false negative cell labels that are missed. If the weight difference is large, then the low-quality cell labels are filtered out, leading to the desirable result where the GMM can fit one cluster to the putative cell cluster and one to the non-cell cluster. When there are multiple tested thresholds with the same largest weight difference, the lowest threshold is chosen.

## Identify a putative cell cluster using the number of transposase sites in peaks - the first GMM

After filtering out low-quality cell labels, the first GMM is performed on all remaining cell labels using the number of transposase sites in peaks. Two clusters are identified (a putative cell cluster and a non-cell cluster), and a cluster assignment is determined for each cell label. When the putative cell cluster and the non-cell cluster do not have good separation, the cell labels at the lower end of the number of transposase sites in peaks are sometimes assigned to the putative cell cluster. This happens due to a high variance estimated for the putative cell cluster. To better define a putative cell cluster using the number of transposase sites in peaks, the maximum value of the number of transposase sites in peaks from the non-cell cluster is calculated and used as the threshold to identify the putative cell cluster. The cell labels in the putative cell cluster proceed to the next step, the second GMM.

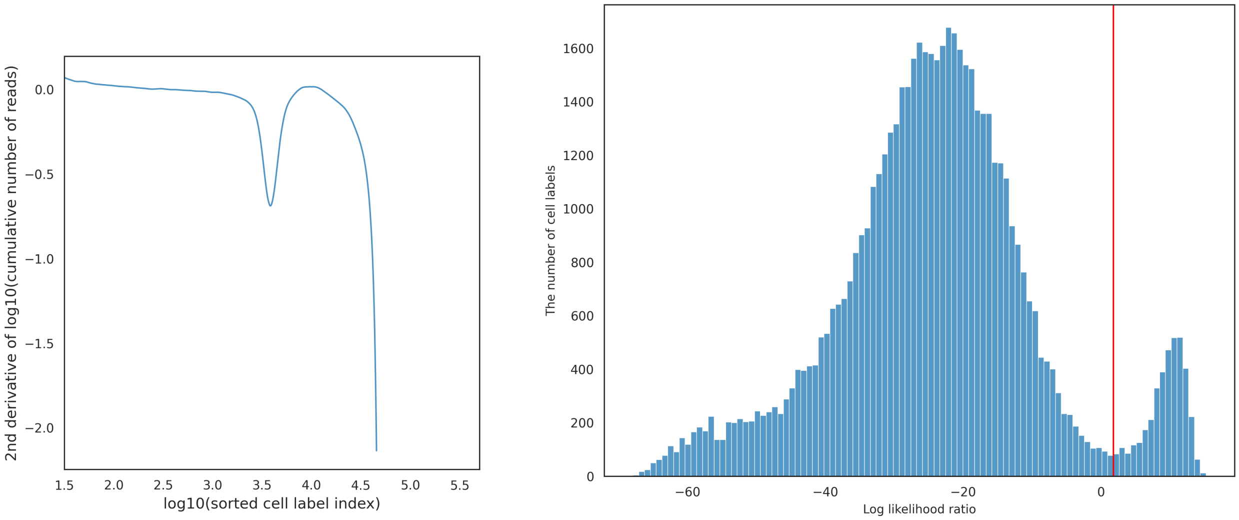
## Identify a putative cell cluster using the number of transposase sites in peaks and fraction of transposase sites in peaks - the second GMM

The cell labels in the putative cell cluster from the first GMM are evaluated using the second GMM, using both the number of transposase sites in peaks and fraction of transposase sites in peaks. The goal of the second GMM is to better define the boundary of the putative cell cluster in the two-dimensional space. The 90% confidence intervals (ellipses) of the two GMM clusters are determined and checked if they overlap. If the two confidence intervals overlap (left plot in the following figure), that indicates a lack of evidence that two distinct clusters exist and thus the second GMM result is ignored. In contrast, when the two intervals do not overlap (right plot in the following figure), that suggests that two distinct clusters exist and the boundary of the putative cell cluster is redefined; the cluster with a higher mean value for the number of transposase sites in peaks is considered a redefined putative cell cluster, and the cell labels assigned to that redefined putative cell cluster are identified as putative cells.

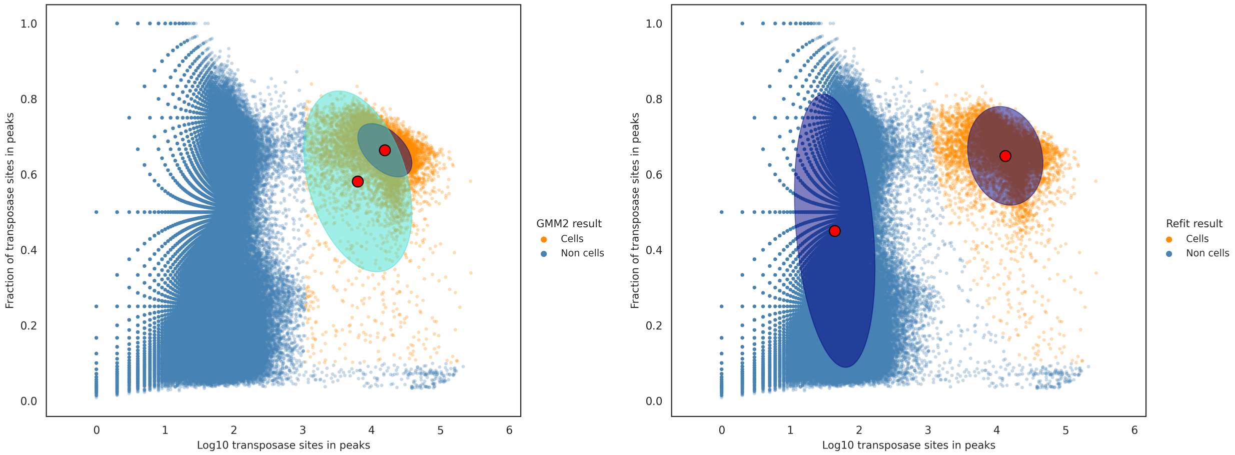


## Refine the boundary of the putative cell cluster

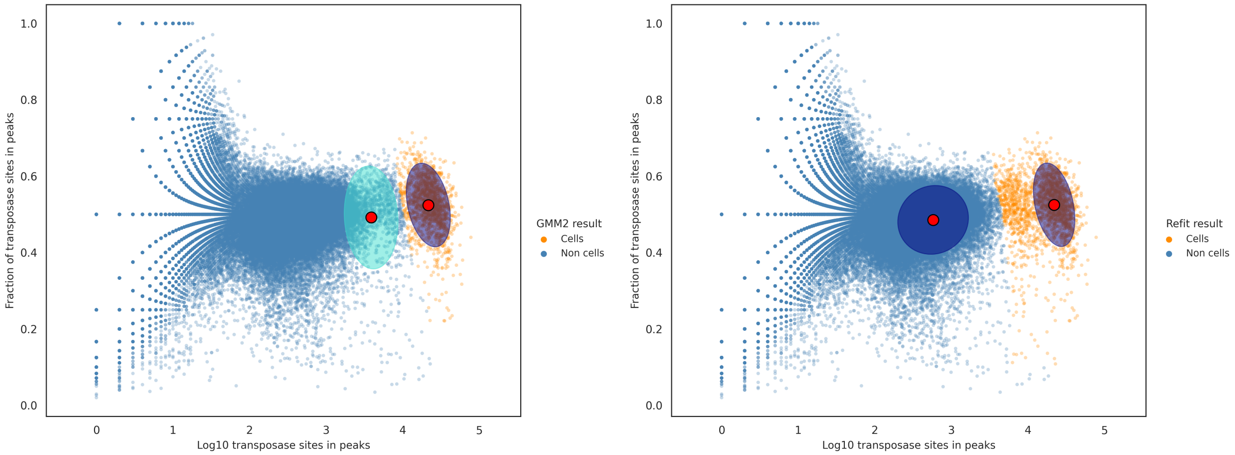
This final refitting step is aimed at improving the boundary identification of the putative cell cluster, either by removing false positives or recovering false negatives. If the previous second GMM result is ignored, the putative cell calling results could contain a small amount of false positives. If the previous second GMM result is applied, the putative cell calling results could be missing a small amount of false negatives. To further refine the boundary of the putative cell cluster in the two-dimensional space, the non-cell labels and the putative cell labels are fitted into two separate two-dimensional Gaussian distributions. Here, the non-cell labels are the cell labels filtered out from the first and second GMM, not including low-quality cell labels. Using the two resulting Gaussian distributions, a log likelihood ratio of the Gaussian distribution from the putative cell labels to the Gaussian distribution from the non-cell labels is calculated for each cell label. The threshold of the log likelihood ratio is dynamically determined for each dataset using the Basic algorithm for putative cell identification using second derivative analysis (page 41), but instead of using reads or transposase sites in peaks, the log likelihood ratios are used. Also, here, the half depth criteria is not used to select the inflection point. The following figure shows the second derivative curve (left) and the histogram of log likelihood ratios (right) with the red line being the selected threshold from the basic algorithm.



When the result from the second GMM in the previous step is ignored, the cell calling results could contain a small amount of false positive cell labels. The false positive cell labels are defined as cell labels having a log likelihood ratio less than the dynamically determined threshold from above, and are filtered out in this step. The following figure shows the putative cell and non-cell clusters before the false positives are removed (left) and after (right).



In contrast, when the result from the second GMM in the previous step is used to filter out non-cell labels, the cell calling results could be missing a small amout of false negative cell labels. The false negative cell labels are defined as cell labels having a log likelihood ratio greater than or equal to the dynamically determined threshold from above, and are recovered in this step. The following figure shows the putative cell and non-cell clusters before the false negatives are recovered (left) and after (right).



This step improves the robustness of the putative cell calling result, especially when the result from the second GMM in the previous step had a minimal overlap or a minimal gap between the two confidence intervals. This improved robustness comes from the log odds ratio threshold being dynamically chosen. If there was a minimal overlap in the confidence intervals from the second GMM, then the second GMM result was ignored, and the log likelihood ratio threshold chosen here would remove false positive cell labels. If there was a minimal gap between the two confidence intervals from the second GMM, then the second GMM result was used to filter out non-cell labels, and the log likelihood ratio threshold chosen here would recover false negative cell labels.

## Identify Protein Aggregates from AbSeq Read Counts

When identifying putative cells using the AbSeq read counts, the basic implementation may include a small number of false positive cell labels due to protein aggregates. Putative cells identified with high expression across most AbSeq targets are considered protein aggregates. The protein aggregate status for each putative cell can be found in the `<sample_name>_Protein_Aggregates_Experimental.csv` file. The cell label is marked True if it is considered a potential protein aggregate and False if not.

# Sample Tag Analysis

## Sample multiplexing option

Multiple samples of cell suspension can be loaded into a BD Rhapsody™ cartridge using a BD® Single-Cell Multiplexing Kit. Each sample is labeled with a separate Sample Tag from the kit. The human and mouse sample kits provide up to 12 species-specific sample tags. The flex sample kit provides up to 24 species and cell type agnostic sample tags.

When you start the BD Rhapsody™ Sequence Analysis Pipeline, you can select the sample multiplex option. You can associate a name with a Sample Tag before the pipeline starts, and the specified sample names will be used in the output files.

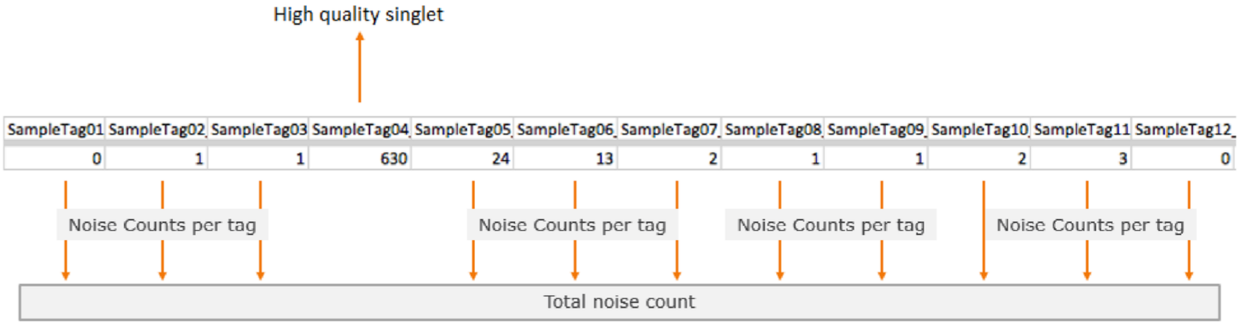
To account for every Sample Tag, each Sample Tag sequence in the kit is considered during pipeline analysis, whether the Sample Tags are used in the experiment or specified with a sample name.

The pipeline automatically adds the Sample Tag barcode sequences to the appropriate reference input files. Reads that align to a Sample Tag sequence and associate with a putative cell are used to identify the sample for that cell.

## Sample determination algorithm

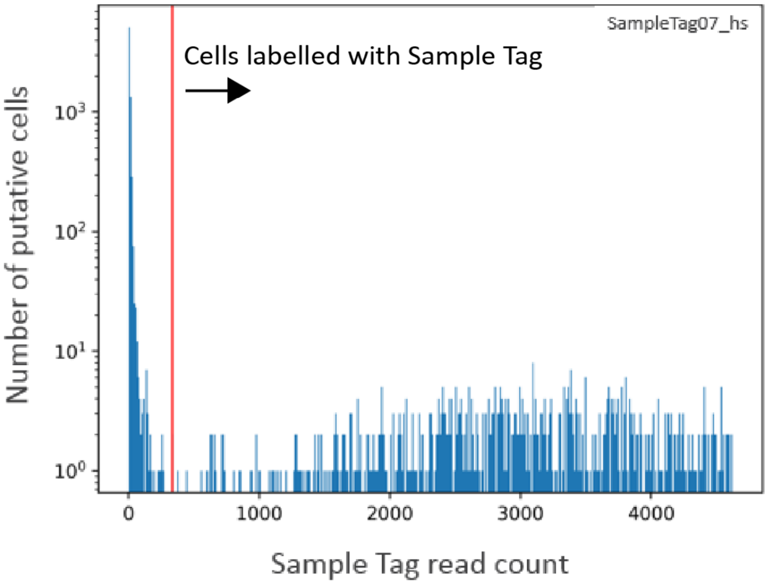
The algorithm first identifies high quality singlets. A high quality singlet is a putative cell where more than 75% of Sample Tag reads are from a single tag. When a singlet is identified, the counts for all the other tags are considered Sample Tag noise. Sources of low-level noise can be: barcode contimination from the oligo manufacturing process, or incomplete washing of individual cell samples, resulting in residual Sample Tag labeling during cell preparation and cartridge steps.

The following image shows an example of Sample Tag read counts for an individual putative cell that is considered a high quality singlet, labeled SampleTag04. All of the other Sample Tag counts are recorded as separate noise counts and are summed to find the noise read count for that putative cell.

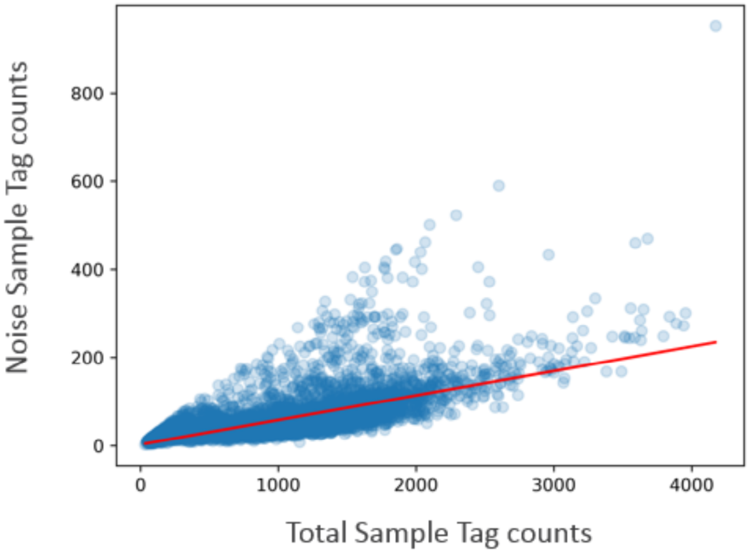


| SampleTag01 | SampleTag02 | SampleTag03 | SampleTag04 | SampleTag05 | SampleTag06 | SampleTag07 | SampleTag08 | SampleTag09 | SampleTag10 | SampleTag11 | SampleTag12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 630 | 24 | 13 | 2 | 1 | 1 | 2 | 3 | 0 |

The minimum Sample Tag read count for a putative cell to be positively identified with a Sample Tag is defined as the lowest read count of a high quality singlet for that Sample Tag. The following histogram shows the of number of Sample Tag reads per putative cell for one of the Sample Tags. The red vertical line indicates the threshold of minimum Sample Tag read count. Putative cells with Sample Tag read counts greater than the threshold (to the right of the red line) are considered labelled with this Sample Tag. In addition to singlets, these putative cells can include multiplets, which are cell labels associated with more than one Sample Tag.



The percentage of noise contribution of each Sample Tag for all cells is calculated by dividing the total per tag noise by the total overall noise. In addition, the total amount of noise versus the total Sample Tag count per putative cell is recorded so that a trend line can be established to estimate the total per-cell noise given an observed number of total Sample Tag count for a cell. The following figure shows an overall noise profile where each dot is a cell. A trend line (in red) is fitted and used to establish the expected amount of noise given a total Sample Tag count. Cells that are off the trend line are likely multiplets. The level of antigen expression across cells can vary, contributing to variation in Sample Tag count per cell. Generally, cells with higher total Sample Tag counts have higher noise Sample Tag counts.



To improve sample determination and recover singlets that are not initially considered high quality, the algorithm subtracts the expected number of per-cell noise counts from each Sample Tag. The total expected per-cell noise, derived from the trend line, is multiplied by the percentage of noise contribution of each Sample Tag to determine the expected noise per Sample Tag.

After subtracting the expected per tag noise, any Sample Tag that has a count higher than its minimum read count is called for that cell, and the putative cell is considered a *called* cell.

When the counts of two or more Sample Tags exceed their minimum thresholds, then that putative cell is called as a cross-sample *Multiplet*, indicating more than one actual cell in the microwell, and the cells are of different samples of origin. Some putative cells might not have enough Sample Tag counts to definitively call their sample of origin, and those are labeled as *Undetermined*.

## Reporting sample origin

If you chose the sample multiplexing option, the main top-level RSEC and DBEC data tables contain counts for putative cells from all samples combined. The sample of origin for each putative cell is listed in the file `<sample_name>_Sample_Tag_Calls.csv`. This file can be used to annotate the combined data tables. The file `<sample_name>_Sample_Tag_Metrics.csv` reports the metrics from the sample determination algorithm. Data tables and metric summary for each sample are output in folders contained in `<sample_name>_Sample_Tag<number>.zip`.

# Generate Expression Matrix Data Tables

RSEC-adjusted molecule counts and DBEC-adjusted molecule counts for each putative cell are presented in matrix market exchange format (MEX). RSEC molecule counts per cell are also preloaded, along with cell metadata, into popular analysis package formats:

- Seurat (.rds)
- Scanpy / Muon (.h5mu)

In addition, an unfiltered expression matrix is output in matrix market exchange format (MEX) for all cell labels that had at least 10 reads associated with them.

During this step, two dimensionality reduction methods are run on the expression data to generate tSNE and UMAP coordinates of the putative cells.

# Annotate BAM

If enabled, the BAM file output by STAR and bwa-mem2 (for ATAC-Seq) is further annotated to summarize the results of the BD Rhapsody™ Sequence Analysis Pipeline. See BAM and BAM Index (page 81) in the output files section for more detail on tags added to BAM records.

See also:

- samtools.github.io/hts-specs/SAMv1.pdf
- STARmanual.pdf
- bwa-mem2 man page (note that usage is same as original bwa-mem)

# TCR and BCR Analysis

## TCR and BCR overview

In combination with other BD Rhapsody™ assays, optional protocols and products enable the generation of sequencing libraries specific to T- and B-Cell Receptors (TCR and BCR). When enabled, the BD Rhapsody™ Sequence Analysis Pipeline can use the reads from these libraries to assemble contigs corresponding to rearranged TCR and BCR chain mRNA. These contigs are then analyzed to identify single-cell level VDJ gene segments, complementary determining regions (CDRs), read and molecule counts, and per cell type chain-pairing. TCR and BCR analysis is supported for human and mouse.

## Major TCR and BCR pipeline steps

1. Identify reads derived from TCR or BCR mRNA (page 57)
2. Assemble reads into contigs (page 57)
3. Annotate contigs with VDJ gene segment information (page 58)
4. Select dominant contigs and chain family (page 58)
5. Cell analysis - type and quality (page 59)
6. Error correction and contig trimming (page 59)

## Identify reads derived from TCR or BCR mRNA

As described in earlier steps, reads are aligned against a reference sequence to determine their biotype and identity (for example, which gene, AbSeq, sample tag, or VDJ gene segment).

For the WTA assay, in combination with TCR and BCR, the pipeline will identify TCR or BCR reads which align to known VDJ gene segments in the transcriptome, with the appropriate orientation. Known VDJ segments are those with a transcriptome GTF "gene_biotype" starting with "TR_"or "IG_".

For the targeted assay in combination with TCR and BCR, the pipeline automatically adds species appropriate TCR and BCR gene segments to the FASTA reference file. These gene segments are derived from the same Gencode transcriptome GTF as is used in the pre-built WTA assay reference archives.

Reads that align to TCR or BCR gene segments are grouped and separated from the reads aligning to other biotypes. These reads then only go through the procedures described in the remainder of this section, and not the steps described previously for other biotypes.

## Assemble reads into contigs

To generate the full variable region of a TCR or BCR sequence, or the consensus CDR3 sequence, short reads must be assembled. Read assembly operates by looking for similarities and overlaps between reads that suggest they originate from the same original sequence. Aligning and stitching these reads together can allow for the creation of longer contigs from short reads, and correct randomly distributed sequencing errors.

Reads identified as TCR or BCR derived, are prepared for assembly with trimming and UMI error correction. First, the 3' end of reads are trimmed with a quality score threshold of 20. It's important that the reads going into assembly be of high quality, so that reads can be correctly aligned, and a valid consensus sequence can be generated. Next, reads are also trimmed based on bead capture sequences to remove artifacts from the BD Rhapsody™ cell label sequence. These sequences could interfere with the correct assembly, and may be found at the 3' end of TCR or BCR reads if they were derived from short amplicons. Then, reads undergo UMI error correction, grouped by their cell ID and the TCR or BCR chain type determined by initial alignment (for example, TCR-Alpha, IG-Kappa...). This UMI error correction step uses the same RSEC algorithm described previously.

To begin assembly, reads are grouped by their cell ID and chain type. These read groups are sent through a software package for transcript assembly called Trinity, which generates a list of contig sequences. Then, the reads are aligned back to the newly generated contigs, in order to produce read and molecule counts for each contig. Multiple contigs from each cell represent the rearranged VDJ mRNA sequences, for instance, of TCR Alpha and TCR Beta chains.

## Annotate contigs with VDJ gene segment information

All contigs generated by the assembly step are analyzed to identify V, D, J, and C gene segments, complementarity determining regions (CDR1-3), framework regions (FR1-4), productivity (lack of stop codons), if contig is full length, and protein sequence. This analysis is accomplished with a software package called IGBlast and with alignments using Bowtie2.

A contig is removed from further analysis if a V or J gene selection is of low quality, indicated by an e-value score greater than $10^{-3}$ (lower is better). A contig is considered “full length” when there is amino acid sequence defined for each framework (1-4) and CDR (1-3) region. For the “full length” metric, FR1 and FR4 may be partial, but the overall contig is still considered full length.

All annotated contigs are written to an unfiltered AIRR compliant output file.

## Select dominant contigs and chain family

For each cell and chain type, a dominant contig is selected to facilitate reporting, metrics, and downstream analysis. The selection of a dominant contig follows these rules:

- Read count of the contig is at least 20% of the total read count from all contigs for the cell-chain.
- To break any ties, the contigs are then sorted in order of: productivity, highest molecule count, highest read count, full length, and best V-segment e-value quality score.

Secondary contigs can be generated due to biological reasons, like dual expression of alpha or beta TCR chains expression or assay-based reasons, like: sequencing errors, transcription errors, contaminating reads from other mRNAs, cell multiplets, and misassembly.

Dominant contigs for chain types that do not correspond to the cell type selected (described in the following subsections) are not reported in the dominant contigs output file, but all contigs are available in the unfiltered contigs output file.

For cells expressing both TCR alpha/beta and TCR gamma/delta, a single chain family is selected for the final "per cell" output file, but all data is still available in the uncorrected output. To select the chain family, expression of both alpha and beta or gamma and delta is preferred. Then, if all four chains or one of each chain has expression, the family with the highest combined molecule count is selected (alpha+beta vs gamma+delta).

## Cell analysis - type and quality

During any BD Rhapsody™ assay in combination with TCR and BCR, putative cell determination is still based on 3' gene expression from targeted or WTA data. This is more accurate than creating separate putative cell identifications for the TCR or BCR libraries or both. The VDJ metrics contain a breakdown of metrics by cell type. Cell types are determined in one of two ways. The pipeline contains an experimental immune cell type classifier that uses a series of machine learning models developed on human PBMCs. This method will only work when the targeted or WTA gene expression, or AbSeq expression data tables contain a sufficient number of core genes/AbSeqs relevant to the model.

The TCR and BCR algorithms contain a simple fallback for cell type determination, in the case of human data where gene expression was not available for a sufficient number of core genes, or for mouse data:

- A putative cell with 2x more TCR molecules than BCR molecules, (or only TCR data) is a T cell
- A putative cell with 2x more BCR molecules than TCR molecules, (or only BCR data) is a B cell
- A putative cell without a 2x difference, or one without any TCR or BCR data is unknown

Cell type determination by the fallback mechanism may be different in unfiltered data vs corrected data.

Cells are then further classified according to their quality. High-quality B or T cells are those that match these criteria:

- Called a B or T cell by the above cell type classification algorithm
- Contains at least one productive contig from a cell type appropriate BCR or TCR chain type
- Contains at least four molecules from cell type appropriate BCR or TCR chain type(s)

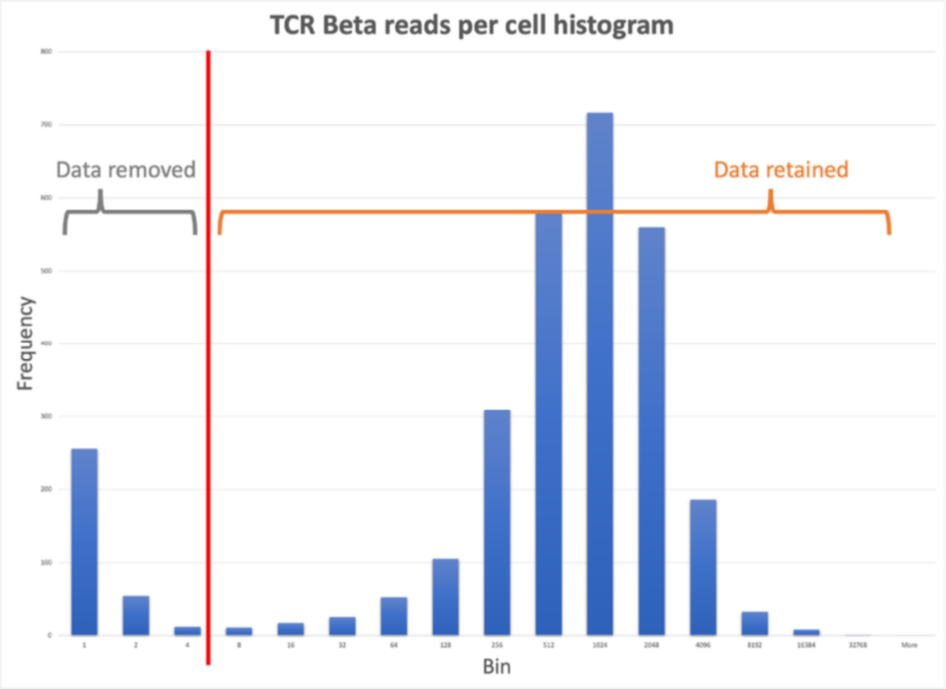
A "high-quality" column in the perCell and contig output files show whether a cell satisfied these criteria. Then, an additional set of per cell VDJ metrics is computed. These high-quality metrics are similar to other single-cell VDJ products where there is a putative cell determination for VDJ libraries alone, separate from the putative cell determination from associated gene expression libraries.

## Error correction and contig trimming

Dominant contigs from putative cells undergo two additional steps before final reporting. First, the 3' end of contigs are trimmed based on the identified constant region and the known primer sequence it contains. Any assembled sequence 3' of the primer sequence, within the constant region, is not consequential to the VDJ region, and possibly assembled in error.

To improve specificity, dominant contigs from putative cells undergo a final round of error correction on a per chain type basis: for each of TCR alpha, beta, gamma, delta, and BCR heavy, kappa, and lambda. This distribution style error correction assumes that each individual chain type from T or B cells will amplify at similar rates, and thus would end up with similar numbers of reads per cell-chain for a real TCR or BCR expressing cell. Artifact molecules created with non-T or non-B cell labels in late PCR cycles, such as those derived from PCR chimera formation, will likely have fewer reads. The algorithm is multi-modal aware, so that if there are 2 positive populations for a particular chain, they should both be kept (for example, Naïve B cells and plasma B cells with different reads per cell in the same experiment).

First, there is a check to determine if each chain type has a read depth of at least four reads per cell. If not, then error correction does not proceed for that chain type. Next, a histogram of the reads per cell from each chain type is generated, and a multi-modal distribution is modeled on each. A threshold is set at the local minima between the first and second modes, and on a per chain basis, any TCR or BCR data from cells whose reads counts are in the lowest mode are removed.



There are two exceptions where contigs will be retained, even if they fall below the threshold for that chain:

1. The cell chain contig has a unique CDR3 that is not seen in other cells
2. The CDR3 paired data matches exactly to another cell that had both chains pass filter (i.e. there is another alpha-beta T cell clone with the same CDR3 pair)

Untrimmed contigs and contigs before error correction are still available in an unfiltered contigs output file.

# ATAC-Seq Analysis

## ATAC-Seq analysis overview

In combination with other BD Rhapsody™ assays, optional protocols and products enable the generation of sequencing single-nuclear ATAC-Seq fragments. When enabled, the BD Rhapsody™ Sequence Analysis Pipeline can use the reads from these libraries to evaluate open chromatin regions. The fragments are then analyzed to identify nucleosome-free regions, peak regions, transcription start site (TSS) enrichment, read and fragment counts, and putative cells. ATAC-Seq analysis is supported for human and mouse.

## Major ATAC-Seq pipeline steps

1. Read Quality Filter (page 61)
2. Annotate I2 Cell Label (page 61)
3. Align R1 and R2 Reads (page 62)
4. Generate Fragments (page 62)
5. Call Peaks (page 62)
6. Generate Cell-by-Peak Matrix (page 62)
7. Putative Cell Calling (page 62)
8. Immune Cell Type Classification (Experimental) (page 62))
9. Dimensionality Reduction (page 63)

## Read Quality Filter

The following filtering criteria are applied to the I2, R1 and R2 reads:

| Minimum I2 Read Length | Minimum Read 1 Length | Minimum Read 2 Length | Minimum Mean Base Quality | I2 Single Nucleotide Frequency | R1 Single Nucleotide Frequency | R2 Single Nucleotide Frequency |
|---|---|---|---|---|---|---|
| 43 | 30 | 30 | 20 | 0.55 | 0.8 | 0.8 |

The filtering is performed identically as previously described in the Filtering criteria (page 29) for processing RNA reads.

## Annotate I2 Cell Label

We utilize the same method as previously described in the Cell label (page 31) section for processing RNA reads.

The bead cell-label sequence is on the I2, and its read structure is as follows:

| Diversity Insert | CLS1 | Linker1 | CLS2 | Linker2 | CLS3 | UMI | Capture Sequence |
|---|---|---|---|---|---|---|---|
| None, A, GT or TCA | 9bp | AATG | 9bp | CCAC | 9bp | NNNNNNNN | GTGGAGTCGTGATTATA |

## Align R1 and R2 Reads

Read fragments that pass quality filters and have a valid cell label on I2 have their R1 and R2 read sequences aligned to a reference using bwa-mem2 (page 127). The bwa-mem2 reference is prebuilt and provided in the Reference_Archive input.

## Generate Fragments

The program sinto is used to process the alignment BAM file and identify the ATAC-Seq genomic fragments indicated by the aligned read pairs. Read pairs that indicate the same genomic fragments are treated as duplicate support for one genomic fragment, because duplication is more likely than two identical fragments in a diploid genome. However, sinto also does differentiate identical fragments that have different cell indexes, reporting the fragment separately for each associated cell index.

## Call Peaks

In order to maximize the amount of information used by MACS2 when identifying regions of the genome with higher than background levels of transposase activity, a BED file containing each end of the fragments identified by sinto is generated and used as input to MACS2. The narrowPeak output of MACS2 is used in this pipeline, for greater precision.

## Generate Cell-by-Peak Matrix

The R package Signac has built-in functionality that takes as input the identified peaks regions and the transposase sites (with associated cell indexes), and calculates the cell-by-peak count matrix. Cell indexes that do not have any associated transposase sites within peak regions are discarded.

## Putative Cell Calling

The ATAC-Seq putative cell-calling algorithms are described in the Determine Putative Cells (page 41) section.

## Immune Cell Type Classification (Experimental)

The pipeline will automatically try to classify immune cell types based on the ATAC-Seq data. If the cells of the experiment are not human PBMCs, this result should be ignored. When putative call calling is performed using only ATAC-Seq data, immune cell types are determined using the ATAC-Seq data (see ATAC-Seq Cell Classification (page 63) in the following subsections). When putative cell calling is done with mRNA and ATAC-Seq data, cell types are determined by jointly using the two datasets (see Joint Cell Classification (page 63) in the following subsections). In both cases, cell classification takes the same approach used in annotating mRNA/Abseq cell types; the approach utilizes a machine learning model trained with human PBMCs and predicts a cell type for each cell index (also described in the TCR and BCR Analysis Cell analysis - type and quality (page 59) subsection). However, both ATAC-Seq and joint cell classification need additional preprocessing steps before applying the machine learning model, as detailed in the following subsections.

### ATAC-Seq Cell Classification

A cell-by-peak matrix is converted into a cell-by-gene matrix using the GeneActivity function in Signac with the default parameters. The GeneActivity function counts the number of fragments overlapping with a promoter region and a gene body of each gene. The resulting ATAC-Seq cell-by-gene matrix is used to predict cell types.

### Joint Cell Classification

When both mRNA and ATAC-Seq data are used to call putative cells, joint cell type classification is performed using an ATAC-Seq cell-by-gene matrix (described in the preceding subtopic), and an mRNA cell-by-genes matrix. The common genes are selected from both matrices. The two matrices are normalized to have the same total counts and then summed up to create a joint cell-by-gene matrix. This resulting joint matrix is used to predict cell types.

## Dimensionality Reduction

To visualize cell indexes in low dimensional space, dimensionality reduction is performed using t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) methods, resulting in two separate sets of coordinates. Preprocessing steps and tSNE/UMAP calculation are both performed using Seurat and Signac. When putative cell calling is done using only ATAC-Seq data, dimensionality reduction is performed on the ATAC-Seq data—see ATAC-Seq Dimensionality Reduction (page 63). When putative cell calling is done with mRNA and ATAC-Seq data, joint dimensionality reduction is performed on the two datasets—see Joint Dimensionality Reduction (page 63). In both cases, when more than 100,000 putative cells exist, the cells are randomly subsampled down to 100,000 cells.

### ATAC-Seq Dimensionality Reduction

The Term Frequency-Inverse Document Frequency (TF-IDF) method is applied to an ATAC-Seq cell-by-peak matrix, and peaks with less than 10 transposase sites in peaks are filtered out. Then Singular Value Decomposition (SVD) is performed, and the first SVD component is filtered out, because it usually has a high correlation with the number of transposase sites in peaks and could interfere with biologically-driven dimensionality reduction and clustering (also described in the Signac vignette). tSNE and UMAP are calculated using the second to 30th component of SVD.

### Joint Dimensionality Reduction

Joint dimensionality reduction is performed using an ATAC-Seq cell-by-peak matrix and an mRNA cell-by-gene matrix. The ATAC-Seq cell-by-peak matrix follows the same preprocessing steps ((TF-IDF, peak filtering, and SVD) as described in the preceding ATAC-Seq Dimensionality Reduction (page 63) subsection. The mRNA cell-by-gene matrix undergoes sctransform normalization and Principal Component Analysis (PCA). The nearest neighbor of each cell index is determined using the FindMultiModalNeighbors function from Seurat, using the second to 30th component of the SVD and the first 50 components of the PCA result. The resulting nearest-neighbor matrix is used to run tSNE/UMAP to get joint dimensionality reduction.

# 4

# Output Files

Most output files contain a header summarizing the pipeline run. Headers contain all of the information needed to rerun the pipeline with the same settings.

## Outputs for all pipeline runs:

| Output | File | Content |
|---|---|---|
| Metrics Summary CSV (page 68) | [sample_name]_Metrics_Summary.csv | Report containing sequencing, molecules, and cell metrics |
| Pipeline Report HTML (page 73) | [sample_name]_Pipeline_Report.html | Summary report containing the results from the sequencing analysis pipeline run |
| Cell-by-Feature Data Tables (page 80)[a, b] | [sample_name]_RSEC_MolsPerCell_MEX.zip<br>[sample_name]_DBEC_MolsPerCell_MEX.zip | Molecules per bioproduct per cell, based on RSEC or DBEC |
| Cell-by-Feature Data Tables (page 80) | [sample_name]_RSEC_MolsPerCell_Unfiltered_MEX.zip | Unfiltered tables containing all cell labels with ≥10 reads |
| BAM and BAM Index (page 81) (if parameter is set) | [sample_name]_Bioproduct.bam<br>[sample_name]_Bioproduct.bam.bai | Alignment file of R2 with associated R1 annotations for Bioproduct |
| Single-Cell Analysis Tool Inputs (page 82) | [sample_name]_Seurat.rds<br>[sample_name].h5mu | Seurat (.rds) input file containing RSEC molecules data table and all cell annotation metadata. Scanpy / Muon input file containing RSEC molecules data table and all cell annotation metadata. |
| Bioproduct Statistics (page 83) | [sample_name]_Bioproduct_Stats.csv | Metrics from RSEC and DBEC Unique Molecular Identifier adjustment algorithms on a per-bioproduct basis |
| Dimensionality Reduction Coordinates (page 100) | [sample_name]_(assay)_tSNE_coordinates.csv<br>[sample_name]_(assay)_UMAP_coordinates.csv | Dimensionality reduction coordinates per cell index |

| Output | File | Content |
|---|---|---|
| Immune Cell Classification Result (page 99) | [sample_name]_(assay)_cell_type_experimental.csv | Immune cell type prediction per cell index |

(a) For a multiplexed samples run, theses tables contain counts for putative cells from all samples combined.
(b) DBEC data table is only output if the experiment includes targeted mRNA or AbSeq bioproducts

## Outputs when multiplex option selected:

| Output | File | Content |
|---|---|---|
| Sample Tag Metrics (page 85) | [sample_name]_Sample_Tag_Metrics.csv | Metrics from the sample determination algorithm |
| Sample Tag Calls (page 86) | [sample_name]_Sample_Tag_Calls.csv | Assigned Sample Tag for each putative cell |
| Sample Tag Folders (page 87) | [sample_name]_Sample_Tag[number].zip<br>[sample_name]_Multiplet_and_Undetermined.zip | Separate data tables and metric summary for cells assigned to each sample tag. **Note:** For putative cells that could not be assigned a specific Sample Tag, a Multiplet_and_Undetermined.zip file is also output. |

## Outputs when VDJ option selected:

| Output | File | Content |
|---|---|---|
| VDJ Metrics (page 88) | [sample_name]_VDJ_Metrics.csv | Overall metrics from the VDJ analysis |
| VDJ Per Cell (page 91) | [sample_name]_VDJ_perCell.csv<br>[sample_name]_VDJ_perCell_uncorrected.csv | Cell specific read and molecule counts, VDJ gene segments, CDR3 sequences, paired chains, and cell type |
| VDJ Dominant Contigs AIRR (page 93) | [sample_name]_VDJ_Dominant_Contigs_AIRR.csv | Dominant contig for each cell label chain type combination (putative cells only) |
| VDJ Unfiltered Contigs AIRR (page 96) | [sample_name]_VDJ_Unfiltered_Contigs_AIRR.csv | All contigs that were assembled and annotated successfully (all cells) |

## Outputs for ATAC-Seq:

| Output | File | Content |
|---|---|---|
| ATAC Metrics (page 105) | [sample_name]_ATAC_Metrics.csv<br>[sample_name]_ATAC_Metrics.json | Overall metrics from the ATAC-Seq analysis |
| ATAC Fragments and Fragments Index (page 101) | [sample_name]_ATAC_Fragments.bed.gz<br>[sample_name]_ATAC_Fragments.bed.gz.tbi | Chromosomal location, cell index, and read support for each fragment detected |
| ATAC Transposase Sites and Index file (page 102) | [sample_name]_ATAC_Transposase_Sites.bed.gz<br>[sample_name]_ATAC_Transposase_Sites.bed.gz.tbi | Chromosomal location, cell index, and read support for each transposase site detected |
| ATAC Peaks and Peaks Index (page 103) | [sample_name]_ATAC_Peaks.bed.gz<br>[sample_name]_ATAC_Peaks.bed.gz.tbi | Peak regions of transposase activity |
| ATAC Cell-by-Peak Data Tables (page 104) | [sample_name]_ATAC_Cell_by_Peak_MEX.zip | Peak regions of transposase activity per cell |
| ATAC Cell-by-Peak Data Tables (page 104) | [sample_name]_ATAC_Cell_by_Peak_Unfiltered_MEX.zip | Unfiltered file containing all cell labels with >=1 transposase sites in peaks |
| BAM and BAM Index (page 81) (if parameter is set) | [sample_name]_ATAC.bam<br>[sample_name]_ATAC.bam.bai | Alignment file for R1 and R2 with associated I2 annotations for ATAC-Seq |

## Output when "Cell_Calling_Data: AbSeq" is selected:

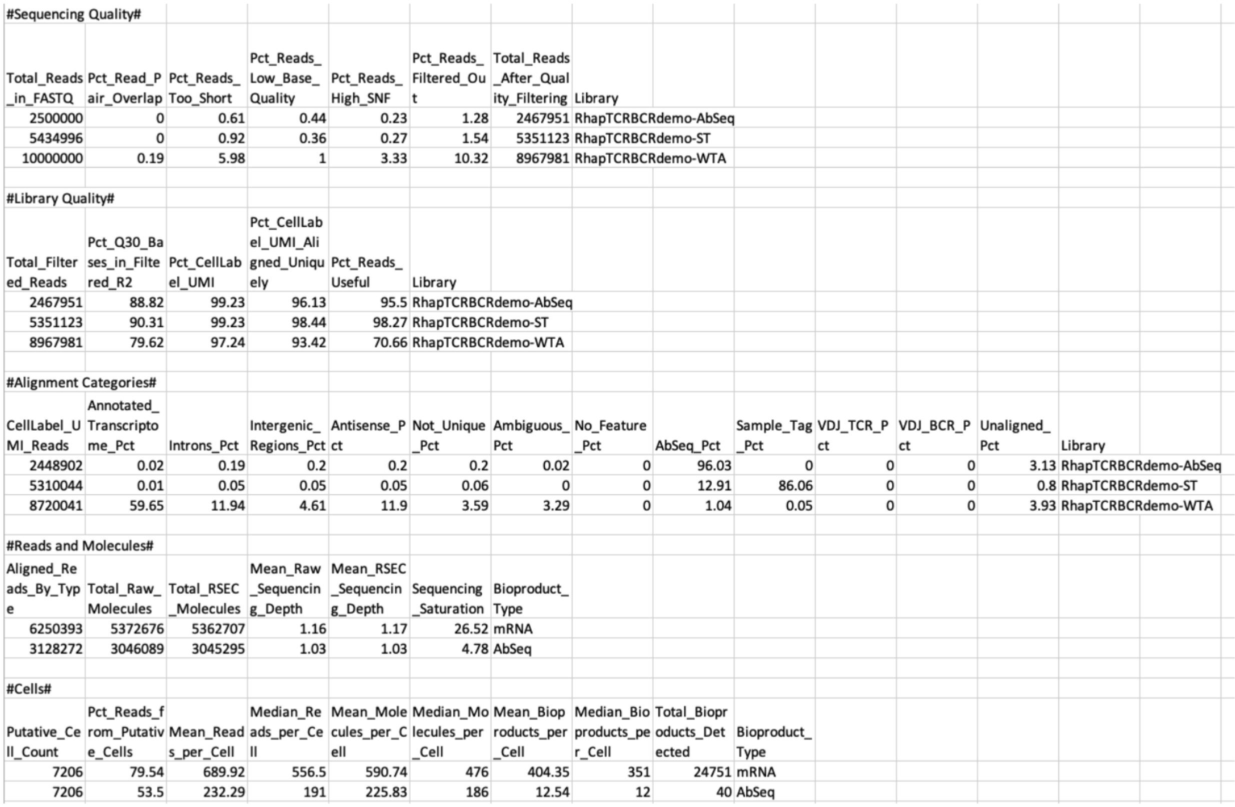
| Output | File | Content |
|---|---|---|
| Protein Aggregates Experimental (page 98) | [sample_name]_Protein_Aggregates_Experimental.csv | Putative cell annotation showing cells suspected of resulting from protein aggregates |

# Metrics Summary CSV

**File**: `[sample_name]_Metrics_Summary.csv`

The Metrics summary provides statistics on sequencing, molecules, cells, and bioproducts.

Sample Tag, VDJ and AbSeq metrics display only when they are used in an experiment.



#Sequencing Quality#

| Total_Reads_in_FASTQ | Pct_Read_Pair_Overlap | Pct_Reads_Too_Short | Pct_Reads_Low_Base_Quality | Pct_Reads_High_SNF | Pct_Reads_Filtered_Out | Total_Reads_After_Quality_Filtering | Library |
|---|---|---|---|---|---|---|---|
| 2500000 | 0 | 0.61 | 0.44 | 0.23 | 1.28 | 2467951 | RhapTCRBCRdemo-AbSeq |
| 5434996 | 0 | 0.92 | 0.36 | 0.27 | 1.54 | 5351123 | RhapTCRBCRdemo-ST |
| 10000000 | 0.19 | 5.98 | 1 | 3.33 | 10.32 | 8967981 | RhapTCRBCRdemo-WTA |

#Library Quality#

| Total_Filtered_Reads | Pct_Q30_Bases_in_Filtered_R2 | Pct_CellLabel_UMI | Pct_CellLabel_UMI_Aligned_Uniquely | Pct_Reads_Useful | Library |
|---|---|---|---|---|---|
| 2467951 | 88.82 | 99.23 | 96.13 | 95.5 | RhapTCRBCRdemo-AbSeq |
| 5351123 | 90.31 | 99.23 | 98.44 | 98.27 | RhapTCRBCRdemo-ST |
| 8967981 | 79.62 | 97.24 | 93.42 | 70.66 | RhapTCRBCRdemo-WTA |

#Alignment Categories#

| CellLabel_UMI_Reads | Annotated_Transcriptome_Pct | Introns_Pct | Intergenic_Regions_Pct | Antisense_Pct | Not_Unique_Pct | Ambiguous_Pct | No_Feature_Pct | AbSeq_Pct | Sample_Tag_Pct | VDJ_TCR_Pct | VDJ_BCR_Pct | Unaligned_Pct | Library |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2448902 | 0.02 | 0.19 | 0.2 | 0.2 | 0.2 | 0.02 | 0 | 96.03 | 0 | 0 | 0 | 3.13 | RhapTCRBCRdemo-AbSeq |
| 5310044 | 0.01 | 0.05 | 0.05 | 0.05 | 0.06 | 0 | 0 | 12.91 | 86.06 | 0 | 0 | 0.8 | RhapTCRBCRdemo-ST |
| 8720041 | 59.65 | 11.94 | 4.61 | 11.9 | 3.59 | 3.29 | 0 | 1.04 | 0.05 | 0 | 0 | 3.93 | RhapTCRBCRdemo-WTA |

#Reads and Molecules#

| Aligned_Reads_By_Type | Total_Raw_Molecules | Total_RSEC_Molecules | Mean_Raw_Sequencing_Depth | Mean_RSEC_Sequencing_Depth | Sequencing_Saturation | Bioproduct_Type |
|---|---|---|---|---|---|---|
| 6250393 | 5372676 | 5362707 | 1.16 | 1.17 | 26.52 | mRNA |
| 3128272 | 3046089 | 3045295 | 1.03 | 1.03 | 4.78 | AbSeq |

#Cells#

| Putative_Cell_Count | Pct_Reads_from_Putative_Cells | Mean_Reads_per_Cell | Median_Reads_per_Cell | Mean_Molecules_per_Cell | Median_Molecules_per_Cell | Mean_Bioproducts_per_Cell | Median_Bioproducts_per_Cell | Total_Bioproducts_Detected | Bioproduct_Type |
|---|---|---|---|---|---|---|---|---|---|
| 7206 | 79.54 | 689.92 | 556.5 | 590.74 | 476 | 404.35 | 351 | 24751 | mRNA |
| 7206 | 53.5 | 232.29 | 191 | 225.83 | 186 | 12.54 | 12 | 40 | AbSeq |

## Metrics summary output sections and metrics

### Sequencing Quality

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_Reads_in_FASTQ | Number of read pairs in the input FASTQ files | Sequencing amount |
| Pct_Read_Pair_Overlap | Percentage of read pairs overlapped with each other | Library quality |
| Pct_Reads_Too_Short | Percentage of read pairs filtered out due to length | Sequencing quality |
| Pct_Reads_Low_Base_Quality | Percentage of reads filtered out due to average base quality score <20 | Sequencing quality |

| Metric | Definition | Major contributing factors |
|---|---|---|
| Pct_Reads_High_SNF | Percentage of read pairs filtered out due to single nucleotide frequency ≥55% for R1 or ≥80% for R2 | Sequencing quality |
| Pct_Reads_Filtered_Out | Percentage of reads removed by the combination of length, quality, and SNF filters | Sequencing quality |
| Library | Name of library | Name of library |

## Library Quality

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_Filtered_Reads | Number of read pairs after length, quality, and SNF filtering | Sequencing amount<br>Sequencing quality<br>Library quality |
| Pct_Q30_Bases_in_Filtered_R2 | Percentage of R2 bases with quality score >30, averaged across all read pairs retained after quality filtering | Sequencing quality |
| Pct_CellLabel_UMI | Percentage of read pairs containing a valid cell label and UMI | Sequencing quality<br>Library quality |
| Pct_CellLabel_UMI_Aligned_Uniquely | Percentage of read pairs containing a valid cell label and UMI that aligned uniquely | Sequencing quality<br>Library quality |
| Pct_Reads_Useful | Percentage of read pairs containing a valid cell label and UMI that aligned uniquely to a valid bioproduct | Sequencing quality<br>Library quality |
| Library | Name of library | Name of library |

## Alignment Categories

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_CellLabel_UMI_Reads | Number of read pairs containing a valid cell label and UMI | Sequencing quality<br>Library quality |
| Annotated_Transcriptome_Pct | Percentage of cellular reads with read 2 aligned uniquely to gene present in the transcriptome (WTA) or mRNA panel (Targeted) | Sequencing quality<br>Library quality<br>Cell type |
| Introns_Pct | Percentage of cellular reads with read 2 aligned uniquely to an intronic region of a gene | Sequencing quality<br>Library quality<br>Cell type |

| Metric | Definition | Major contributing factors |
|---|---|---|
| Intergenic_Regions_Pct | Percentage of cellular reads with read 2 aligned uniquely to an intergenic region | Sequencing quality<br>Library quality<br>Cell type |
| Antisense_Pct | Percentage of cellular reads with read 2 aligned uniquely to an antisense strand | Sequencing quality<br>Library quality<br>Cell type |
| Not_Unique_Pct | Percentage of cellular reads with read 2 not uniquely aligned | Sequencing quality<br>Library quality<br>Cell type |
| Ambiguous_Pct | Percentage of cellular reads with read 2 aligned to region with ambiguous annotation | Sequencing quality<br>Library quality<br>Cell type |
| No_Feature_Pct | Percentage of cellular reads with read 2 aligned to non-annotated region (WTA) or targeted mRNA reads that are filtered out due to an invalid alignment | Sequencing quality<br>Library quality<br>Cell type |
| AbSeq_Pct | Percentage of cellular reads with read 2 aligned to AbSeq reference | Sequencing quality<br>Library quality |
| Sample_Tag_Pct | Percentage of cellular reads with read 2 aligned to Sample Tag | Sequencing quality<br>Library quality |
| VDJ_TCR_Pct | Percentage of cellular reads with read 2 aligned to TCR gene segments | Sequencing quality<br>Library quality |
| VDJ_BCR_Pct | Percentage of cellular reads with read 2 aligned to BCR gene segments | Sequencing quality<br>Library quality |
| Unaligned_Pct | Percentage of cellular reads with read 2 unaligned to reference | Sequencing quality<br>Library quality |
| Library | Name of library | Name of library |

## Reads and Molecules

| Metric | Definition | Major contributing factors |
|---|---|---|
| Aligned_Reads_By_Type | Number of filtered read pairs aligned to bioproduct type | Sequencing quality<br>Library quality |
| Total_Raw_Molecules | Total number of molecules as defined by the unique combination of cell label, bioproduct identity, and UMI | Sequencing depth<br>Library quality |
| Total_RSEC_Molecules | Total number of molecules detected after the RSEC molecular identifier adjustment algorithm | Sequencing depth<br>Library quality |

| Metric | Definition | Major contributing factors |
|---|---|---|
| Mean_Raw_Sequencing_Depth | Average number of read pairs per molecule before molecular identifier adjustment algorithms | Sequencing depth |
| Mean_RSEC_Sequencing_Depth | Average number of read pairs per molecule after the RSEC molecular identifier adjustment algorithm | Sequencing depth |
| Sequencing_Saturation | Percentage of read pairs representing RSEC-adjusted molecules that are sequenced more than once | Sequencing depth |
| Bioproduct_Type | Type of bioproduct in experiment (mRNA or AbSeq) | Assay type |

## Cells

| Metric | Definition | Major contributing factors |
|---|---|---|
| Putative_Cell_Count | Number of cell labels detected by the cell label filtering algorithm | Number of cells input and captured by cartridge workflow<br>Bead handling |
| Pct_Reads_from_Putative_Cells | Percentage of reads that are assigned to putative cells | Cell viability<br>Cartridge workflow performance<br>Staining and washing (AbSeq) |
| Mean_Reads_per_Cell | Average number of reads representing the molecules detected in each cell | Sequencing depth |
| Median_Reads_per_Cell | Median number of reads representing the molecules detected in each cell | Sequencing depth |
| Mean_Molecules_per_Cell | Average number of molecules detected per cell label | Sequencing depth |
| Median_Molecules_per_Cell | Median number of molecules detected per cell label | Sequencing depth |
| Mean_Bioproducts_per_Cell | Average number of bioproducts detected per cell label | Sequencing depth |
| Median_Bioproducts_per_Cell | Median number of bioproducts detected per cell label | Sequencing depth |
| Total_Bioproducts_Detected | Number of bioproducts detected from all cells | Sequencing depth |
| Bioproduct_Type | Type of bioproduct in experiment (mRNA or AbSeq) | Assay type |

## Sample Tags (if applicable)

| Metric | Definition | Major contributing factors |
|---|---|---|
| Sample_Tag_Filtered_Reads | Number of filtered read pairs aligned to Sample Tags | Sequencing depth |
| ST_Pct_Reads_from_Putative_Cells | Percentage of Sample Tag reads that are assigned to putative cells | Cell viability<br>Sample Tag labeling and wash protocols<br>Cartridge workflow performance<br>Sequencing depth |

## VDJ (if applicable)

- See VDJ Metrics (page 88) output

# Pipeline Report HTML

**File**: `[sample_name]_Pipeline_Report.html`

A pipeline report HTML file is generated and contains the results from the sequencing analysis pipeline. The pipeline report is a stand-alone HTML file that requires no internet connection making it easy to share with collaborators. The pipeline report contains several graphs to help visualize the results. There are also helpful tooltips that describe each specific metric in more detail. The pipeline report also contains the pipeline inputs that were specified for the sequencing analysis, allowing you to reproduce the analysis using the same inputs and settings.

## Summary section

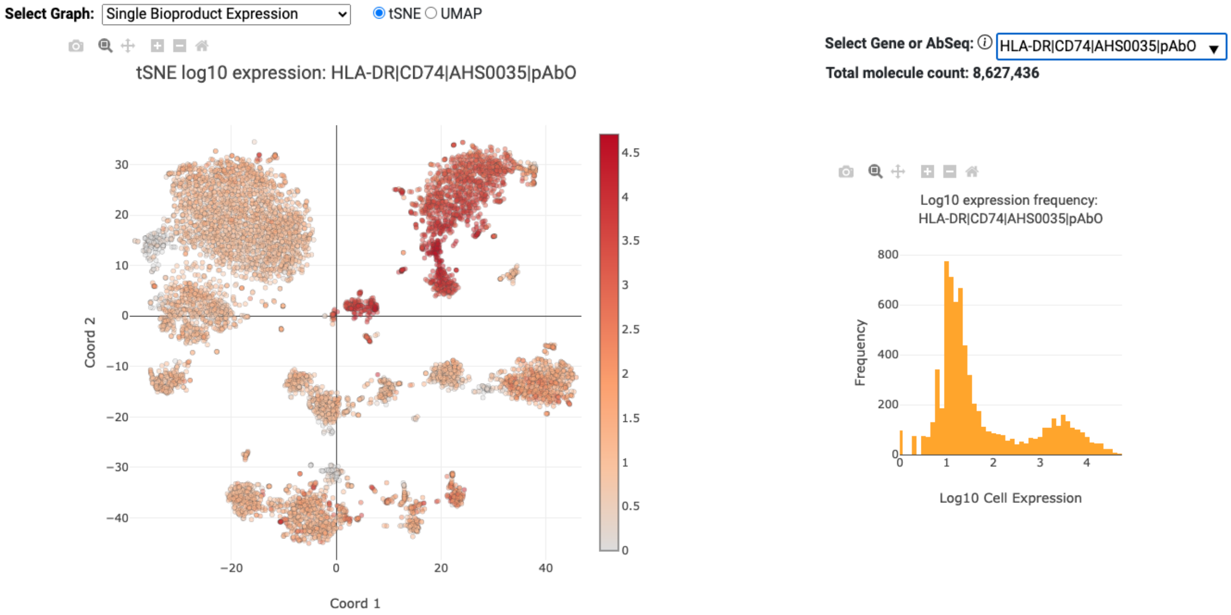
The pipeline report starts out with the summary information at the top with the most important metric results. The number of putative cells is shown at the very top. Underneath the putative cells is a summary table for all bioproducts that were included. On the left side of the bioproducts summary, some key library specific metrics such as the number of reads in the FASTQ, the percentage of reads that passed all the quality filters, and the percentage of reads that had a valid cell label and UMI that aligned uniquely are shown. On the right side of the bioproducts summary, some key bioproduct type metrics such as the number of reads that passed all quality filters that aligned, the average number of reads representing the molcules detected in each cell, and the average number of molecules detected per cell label are highlighted. If ATAC-Seq data was included in the analysis, a summary table for ATAC-Seq metrics will be shown next. Similarly to the bioproduct summary table, the left side of the ATAC-Seq summary will show some key ATAC-Seq library specific metrics. On the right side of the ATAC-Seq summary, some key ATAC-Seq metrics are highlighted.

## Graph section

The graph section has several interactive graphs highlighting some of the most important results from the analysis. For pipeline runs that identify more than 100,000 putative cells, tSNE, UMAP and histogram graphs will show a random sub-sample of 100,000 cells.

### Single Bioproduct Expression (mRNA and AbSeq)

The Single Bioproduct Expression graph displays a tSNE/UMAP on the left and a histogram on the right for individual bioproducts. Each dot on the tSNE/UMAP represents a putative cell and is colored by the log 10 expression of the selected AbSeq target or mRNA gene. The histogram shows the distribution of expression for all cells for the selected AbSeq target or mRNA gene. By default, the bioproduct with the highest expression is selected in the dropdown list. The AbSeq targets and mRNA genes are sorted by total expression (highest to lowest) separately. The sorted AbSeq targets are shown first in the dropdown list followed by the sorted mRNA genes. For experiments with many bioproducts, only the most highly, widely, and variably expressed genes plus all AbSeq targets are shown.



Select Graph: Single Bioproduct Expression

● tSNE ○ UMAP

Select Gene or AbSeq: HLA-DR|CD74|AHS0035|pAbO

Total molecule count: 8,627,436

## Immune Cell Type Experimental (mRNA, AbSeq, and ATAC-Seq)

The Immune Cell Type Experimental graph shows the tSNE/UMAP plot with each cell labeled according to the results from the Cell Type prediction algorithm.

## Total Molecules per cell (mRNA and AbSeq)

The Total Molecules per cell mRNA and AbSeq graphs show the tSNE/UMAP plot on the left where each cell is colored by the log 10 of total expression for all mRNA genes or AbSeq targets respectively. The histogram on the right shows the distribution of total expression for all cells for all mRNA genes or AbSeq targets respectively.

## Total Transposase Sites in Peaks per cell (ATAC-Seq)

The Total Transposase Sites in Peaks per cell ATAC-Seq graph shows the tSNE/UMAP plot on the left where each cell is colored by the log 10 of total transposase sites in peaks. The histogram on the right shows the distribution of total transposase sites in peaks for all cells.

## VDJ BCR/TCR Paired Chains

The VDJ BCR/TCR Paired Chains tSNE/UMAP plots show the clusters of cells with BCR/TCR paired chains.

## Sample Multiplexing

The Sample Multiplexing tSNE/UMAP plot shows the cells labeled by sample tag and includes the multiplet and undetermined cell labels.

## Metric Sections

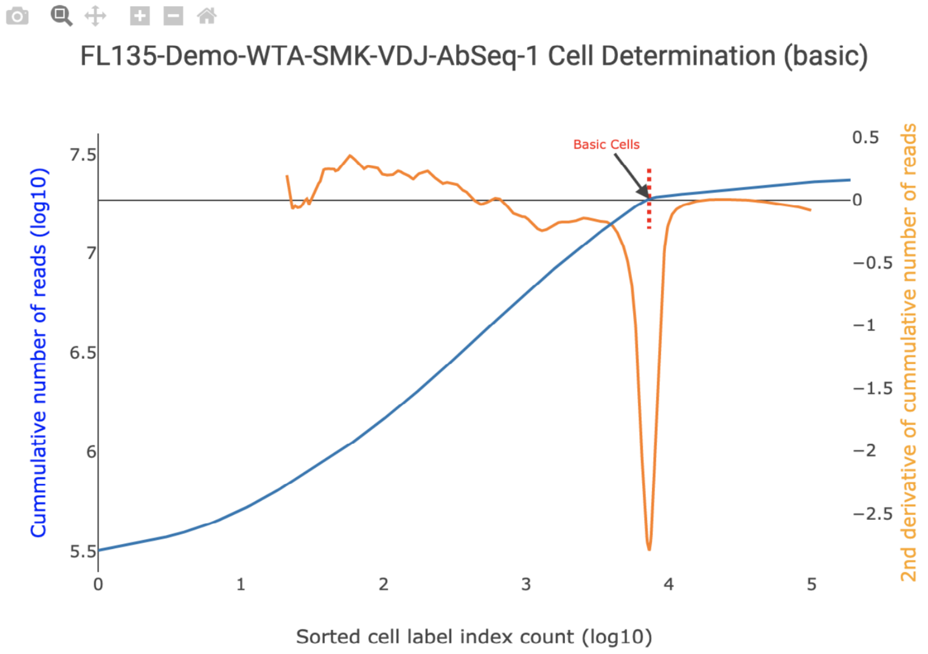
There are several sections in the pipeline report providing details about specific metrics. The main bioproduct sections cover the Sequencing Quality, Library Quality, Alignment Categories, Reads and Molecules, Cell Calling, Sample Multiplexing, and VDJ results. The data in these sections can also be found in the Metrics Summary CSV file. The main ATAC-Seq sections cover Sequencing Quality, Library Quality, Alignment Categories, Fragments, Peaks, and Cell Calling. The data in these sections can also be found in the ATAC Metrics file. More details about some of the sections are provided in the following text..

# Cells Section

The Cells section provides interactive graphs from the basic and refined cell calling algorithms that were described in the Determine Putative Cells (page 41) section. It also includes cell related metrics for all bioproducts and ATAC-Seq data that was included.

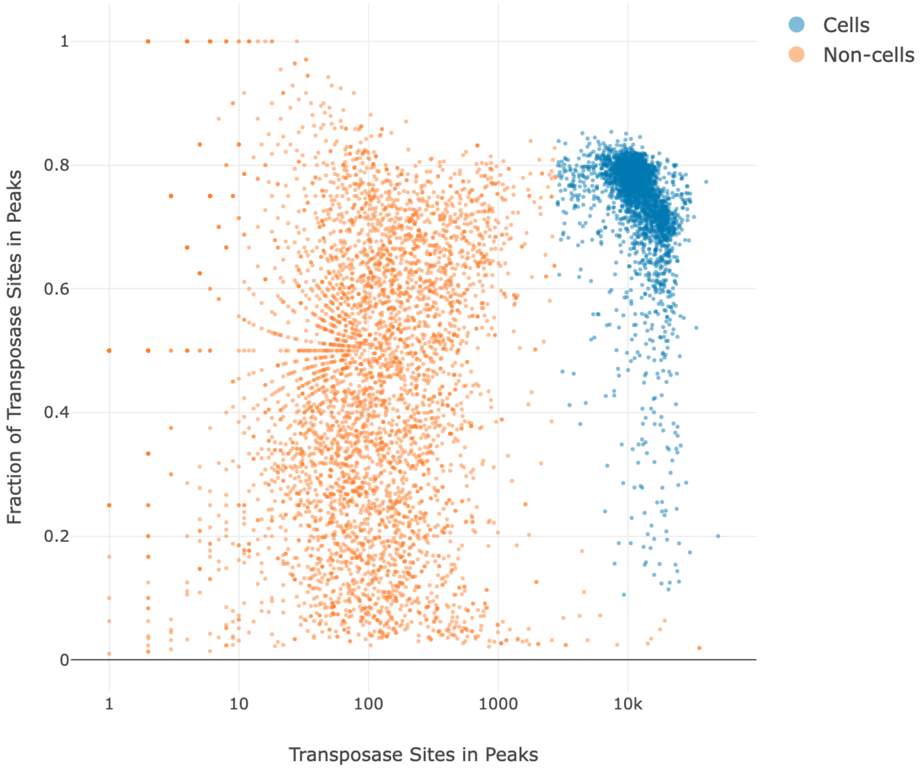
## Bioproduct Cell Calling

By default, the Basic algorithm for putative cell identification using second derivative analysis (page 41) is used for bioproducts. For the basic cell calling algorithm, the second derivative plot is shown on top of the cumulative read plot and the basic cell line is shown in red. Hovering over the graph will display a vertical line that corresponds to the number of putative cells on the cumulative read plot. All general graph functionality is available. See General Graph Functionality (page 78) for details. For pipeline runs set to use the Refined algorithm for adjusting putative cell counts for ATAC-Seq (page 45), the number of false positive, false negative, and low molecule count cells are shown. All together, these show how the final refined cell call number was derived.



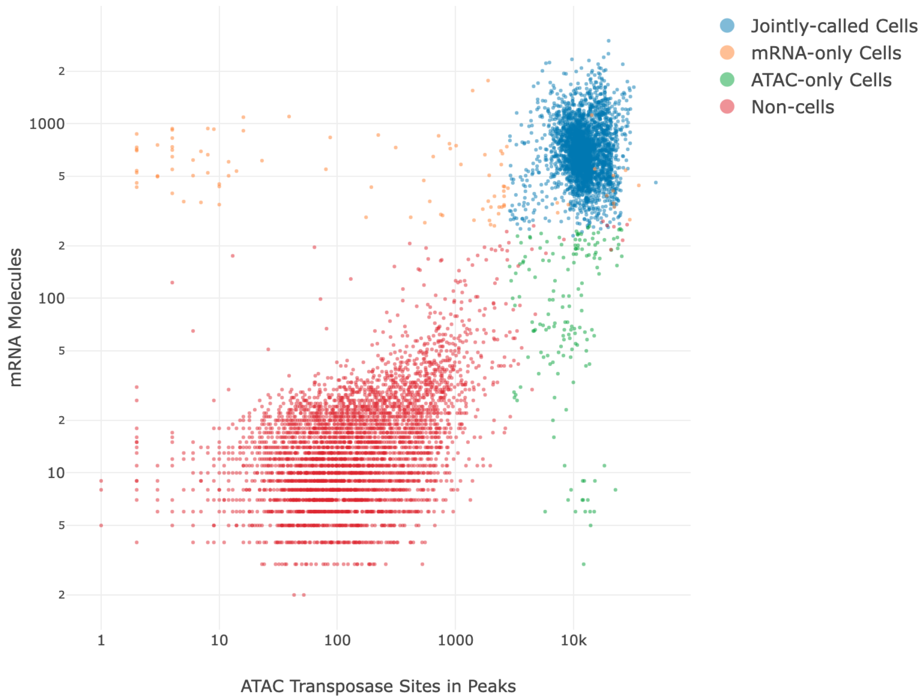
## ATAC-Seq Cell Calling

By default, the Basic algorithm for putative cell identification using second derivative analysis (page 41) is used for ATAC-Seq. For the basic cell calling algorithm, the second derivative plot is shown on top of the cumulative transposase sites in peaks plot and the basic cell line is shown in red. All general graph functionality described in the Bioproduct Cell Calling (page 75) section is available. For pipeline runs set to use the Refined algorithm for adjusting putative cell counts for ATAC-Seq (page 45), the putative cell and non-cell clusters from the Guassian Mixture Model (GMM) refined algorithm are shown on a scatter plot with the number of transposase sites in peaks on the x-axis and the fraction of transposase sites in peaks on the y-axis. See the Determine Putative Cells (page 41) section for more details. The initial number of putative cells determined by the first and second GMMs using both the number of transposase sites in peaks and the fraction of transposase sites in peaks is shown along with the false positive and false negative cell labels determined by the final refitting step.



## Joint mRNA and ATAC-Seq Cell Calling

By default, the basic cell calling algorithm is used for joint mRNA and ATAC-Seq cell calling. The user can select either the basic or refined algorithm for mRNA and ATAC-Seq separately. After the putative cells are called separately, the intersection of the two sets of cells make up the final putative cell count. There are three graphs available for joint mRNA and ATAC-Seq cell calling. For mRNA, the basic cell calling graph will be shown. For ATAC-Seq, either the basic or refined cell calling graph will be shown, depending on the algorithm selected. The joint cell calling plot shows the jointly called cells, the mRNA-only cells, the ATAC-only cells, and the non-cells on a scatter plot with the number of transposase sites in peaks on the x-axis and the number of mRNA UMIs on the y-axis. See the Determine Putative Cells (page 41) section for more details. The number of jointly called cells along with the number of cells detected by each algorithm separately are reported.



# Sample Multiplexing

In the sample multiplexing section, there is summary information such as the number of filtered reads that aligned to the sample tags and the percentage of sample tag reads that are assigned to putative cells. There is also a detailed section showing the number of reads and percentage of reads assigned to each sample tag, along with the number of cells, percentage of cells, number of reads per cell, and mean reads per cell for each sample tag. The detailed section also shows the number of multiplets and undetermined cells.

## VDJ

In the VDJ section, the first table for the "Reads" and the second table for the "Molecules and Dominant Contigs" show the collapsed summary information for the Chain Category (BCR/TCR). By pressing the down arrow, the table expands to show more details about the specific chains. There is also a section for Cell Type specific metrics. There are four tables that can be selected from the dropdown menu: Paired Chains Pct, Pct Cells Positive, Pct Cells Full Length, and Mean Molecules per Cell.

## ATAC-Seq

### Fragment Length Distribution Plot

In the ATAC-Seq Fragments section, the fragment length distribution plot is shown. Chromatin of healthy cells is organized into nucleosomes by wrapping DNA around histone proteins. DNA within a nucleosome is not accessible to Tn5 transposase, so the genomic fragments of ATAC-Seq experiments tend to have a characteristic multi-peaked length distribution. The fragment lengths in the valleys of the distribution correspond to lengths of DNA necessary to be involved in one or more nucleosomes. Fragments under 147 bp in length are classified by BD Rhapsody™ as being in a Nucleosome-Free Region (NFR). Fragments of length between 147 and 294 bp are classified as mononucleosomal length. Lacking the characteristic NFR, mononucleosomal, and dinucleosomal peaks in the Fragment Length Distribution Plot is often an indicator of a low-quality ATAC-Seq experiment, because it suggests the genomic DNA is not organized in nucleosomes (often because of dead/dying cells or ineffective handling of sample nuclei).

### TSS Enrichment Plot

In the ATAC-Seq Peaks section, the TSS enrichment plot is shown. Transcription Start Site (TSS) enrichment is calculated by aggregating the amount of Tn5 activity in the regions around every annotated TSS in the genome. For each TSS region (defined here as the annotated TSS, the 2000 bp upstream of the TSS, and the 2000 bp downstream of the TSS), all the Tn5 sites within the region are identified, along with their distance from the TSS. The total count of Tn5 sites at each distance within the TSS regions are added up, and then the values are normalized by dividing by the background rate of Tn5 activity (defined here as the average number of transposase sites in the first and last 100bp of the 4001 bp TSS region). The TSS Enrichment Score for an experiment is defined as the peak value of the TSS Enrichment Plot.

## Metric Alerts

The Metric Alerts section provides information about metrics from the experiment that are above or below certain thresholds that are typical for most experiments. The alert will specify the library or bioproduct, metric, metric value, threshold, and some possible causes and suggestions.

## General Graph Functionality

There are several ways to interact with the graphs. The toolbar provides the following functionality (from left to right):

| Graph function | Description |
|---|---|
| Download Plot | Allows you to download the plot in SVG format. Once downloaded, the SVG is a static image. |

| Graph function | Description |
|---|---|
| Zoom | Allows you to create a box which will zoom in to show the selected region in the graph area. |
| Pan | Allows you to move the graph to center on a different part of the graph to observe it clearer. |
| + | Zooms in 1 level around the center of the graph. |
| – | Zooms out 1 level around the center of the graph. |
| Home | Resets graph to original zoom and axes. |

Additional graph features:

- Color bar: The color bar on the right side of the Single Bioproduct Expression and Total Molecules per cell (mRNA and AbSeq) and Total Transposase Sites in Peaks per cell (ATAC) tSNE/UMAP plots show the intensity of log 10 based expression or counts.
- Hover: Hovering over the points on the graphs will give extra information (for example: cell index, expression level, or counts).

# Cell-by-Feature Data Tables

Files containing putative cells only:

```
[sample_name]_RSEC_MolsPerCell_MEX.zip
[sample_name]_DBEC_MolsPerCell_MEX.zip
```

Unfiltered file containing all cell indexes with >=10 total reads

```
[sample_name]_RSEC_MolsPerCell_Unfiltered_MEX.zip
```

The number of molecules of each bioproduct from each cell is represented in the matrix market exchange (MEX) format. The MEX format is an efficient way to store sparse data, and is a common input format for many single-cell analysis tools. The MEX.zip output files contain three separate gzip compressed files that together represent the molecule counts of each bioproduct in each cell. By convention, these files are named:

- **barcodes.tsv**: Containing a list of cell indexes (integer between 1 and $384^3$), one per row.
- **features.tsv**: Containing a list of bioproducts detected, one per row. For improved compatibility, this file contains three columns, the first two of which are a duplicated gene symbol or AbSeq, and a third which indicates mRNA (Gene Expression) or AbSeq (Antibody Capture) types.
- **matrix.mtx**: Containing a three column per row representation of molecule counts. First column is the 1-based row number from features.tsv (bioproduct). Second column is the 1-based row number from barcodes.tsv (cell). Third column is the molecule count detected for that bioproduct in that cell.

Cell indexes in the barcodes.tsv file are sorted numerically. Bioproducts in the features.tsv file are sorted alphabetically.

# BAM and BAM Index

**Files**:

```
[sample_name]_Bioproduct.bam
[sample_name]_Bioproduct.bam.bai
[sample_name]_ATAC.bam
[sample_name]_ATAC.bam.bai
```

**Note:**

1. The `*_Bioproduct.bam` consists of reads arising from AbSeq, SampleTag, Targeted, VDJ or WTA assays.
2. A `Combined_` prefix is added to the bam and bai files when SampleTags are present in the experiment to signify that the resulting bam contains reads from all the samples.

BAM is an alignment file in binary format that is generated by the aligner and contains tags related to alignment quality. The Bioproduct BAM consists of alignments from the R2 reads only, whereas the ATAC BAM contains alignments from both the R1 and R2 reads. The BAM files are sorted according to the alignment coordinates of either the R2 read (Bioproduct) or both the R1 and R2 reads (ATAC) on each chromosome. The BAM Index is the index file associated with the coordinate-sorted BAM file.

The BD Rhapsody™ Sequence Analysis Pipeline further annotates the BAM files with the tags described in the following table. For the Bioproduct BAM, if a read has multiple alignments (NH tag > 1), then only the first alignment (HI tag is 0 or 1) will be annotated along with all uniquely aligned reads. For the ATAC BAM, all reads will be annotated with the following tags:

| Tag | Definition |
|---|---|
| CB | A number between 1 and $384^3$ representing a unique cell label sequence (CB = 0 when no cell label sequence is detected). |
| MR | Raw molecular identifier sequence. (Bioproduct BAM only) |
| MA | RSEC-adjusted molecular identifier sequence. If not a true cell, the raw UMI is repeated in this tag. (Bioproduct BAM only) |
| CN | Indicates if a sequence is derived from a putative cell, as determined by the cell label filtering algorithm (*T*: putative cell; *x*: invalid cell label or noise cell).<br>**Note:** You can distinguish between an invalid cell label and a noise cell with the CB tag (invalid cell labels are 0). |
| ST | The value is 1–24, indicating the Sample Tag of the called putative cell, or *M* for multiplet, or *x* for undetermined. |
| XF | Name of the Gene/AbSeq/SampleTag that a particular read was annotated to. (Bioproduct BAM only) |

**Note:** A BAM file can be converted to a tab-delimited text file (SAM format) by using Samtools (see htslib.org)

# Single-Cell Analysis Tool Inputs

Seurat file: `[sample_name]_seurat.rds`

Scanpy / Muon file: `[sample_name].h5mu`

Prebuilt input files to popular third-party single-cell analysis toolkits:

- Seurat 4 (satijalab.org/seurat/)

and

- Scanpy / Muon (scanpy.readthedocs.io) / (muon.scverse.org)

In RNA and AbSeq experiments, each file contains the RSEC molecules-per-cell data table for putative cells, along with cell and bioproduct metadata. In ATAC-Seq experiments, each file contains the cell-by-peak data table for putative cells, along with cell and bioproduct metadata.

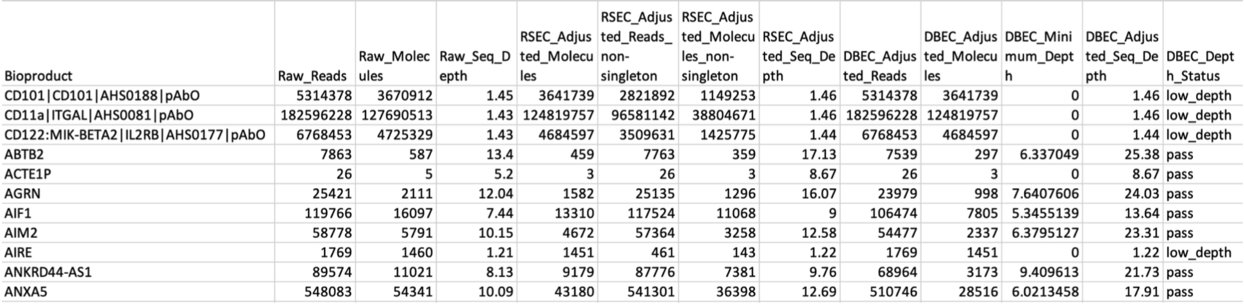
Metadata includes (if applicable): sample tag calls, putative cell origin, TCR/BCR chain types and CDR3, Immune cell type (Experimental), protein aggregate, and tSNE/UMAP coordinates.

For experiments that contain more than one of mRNA, AbSeq, and ATAC-Seq results, these multimodal expression datatypes are built into separate "assay" objects within Seurat ("RNA", "ADT", and "peaks") or MuData ("rna", "prot", and "atac") respective data structures.

# Bioproduct Statistics

**File**: `[sample_name]_Bioproduct_Stats.csv`

The molecular identifier adjustment algorithms RSEC and DBEC are applied to each bioproduct. The molecular identifier metrics file lists the metrics from RSEC and DBEC on a per-bioproduct basis. For more information on RSEC and DBEC molecular identifier adjustment algorithms, see Molecules and Error Correction (page 35). For example:



| Bioproduct | Raw_Reads | Raw_Molecules | Raw_Seq_Depth | RSEC_Adjusted_Molecules | RSEC_Adjusted_Reads_non-singleton | RSEC_Adjusted_Molecules_non-singleton | RSEC_Adjusted_Seq_Depth | DBEC_Adjusted_Reads | DBEC_Adjusted_Molecules | DBEC_Minimum_Depth | DBEC_Adjusted_Seq_Depth | DBEC_Depth_Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CD101\|CD101\|AHS0188\|pAbO | 5314378 | 3670912 | 1.45 | 3641739 | 2821892 | 1149253 | 1.46 | 5314378 | 3641739 | 0 | 1.46 | low_depth |
| CD11a\|ITGAL\|AHS0081\|pAbO | 182596228 | 127690513 | 1.43 | 124819757 | 96581142 | 38804671 | 1.46 | 182596228 | 124819757 | 0 | 1.46 | low_depth |
| CD122:MIK-BETA2\|IL2RB\|AHS0177\|pAbO | 6768453 | 4725329 | 1.43 | 4684597 | 3509631 | 1425775 | 1.44 | 6768453 | 4684597 | 0 | 1.44 | low_depth |
| ABTB2 | 7863 | 587 | 13.4 | 459 | 7763 | 359 | 17.13 | 7539 | 297 | 6.337049 | 25.38 | pass |
| ACTE1P | 26 | 5 | 5.2 | 3 | 26 | 3 | 8.67 | 26 | 3 | 0 | 8.67 | pass |
| AGRN | 25421 | 2111 | 12.04 | 1582 | 25135 | 1296 | 16.07 | 23979 | 998 | 7.6407606 | 24.03 | pass |
| AIF1 | 119766 | 16097 | 7.44 | 13310 | 117524 | 11068 | 9 | 106474 | 7805 | 5.3455139 | 13.64 | pass |
| AIM2 | 58778 | 5791 | 10.15 | 4672 | 57364 | 3258 | 12.58 | 54477 | 2337 | 6.3795127 | 23.31 | pass |
| AIRE | 1769 | 1460 | 1.21 | 1451 | 461 | 143 | 1.22 | 1769 | 1451 | 0 | 1.22 | low_depth |
| ANKRD44-AS1 | 89574 | 11021 | 8.13 | 9179 | 87776 | 7381 | 9.76 | 68964 | 3173 | 9.409613 | 21.73 | pass |
| ANXA5 | 548083 | 54341 | 10.09 | 43180 | 541301 | 36398 | 12.69 | 510746 | 28516 | 6.0213458 | 17.91 | pass |

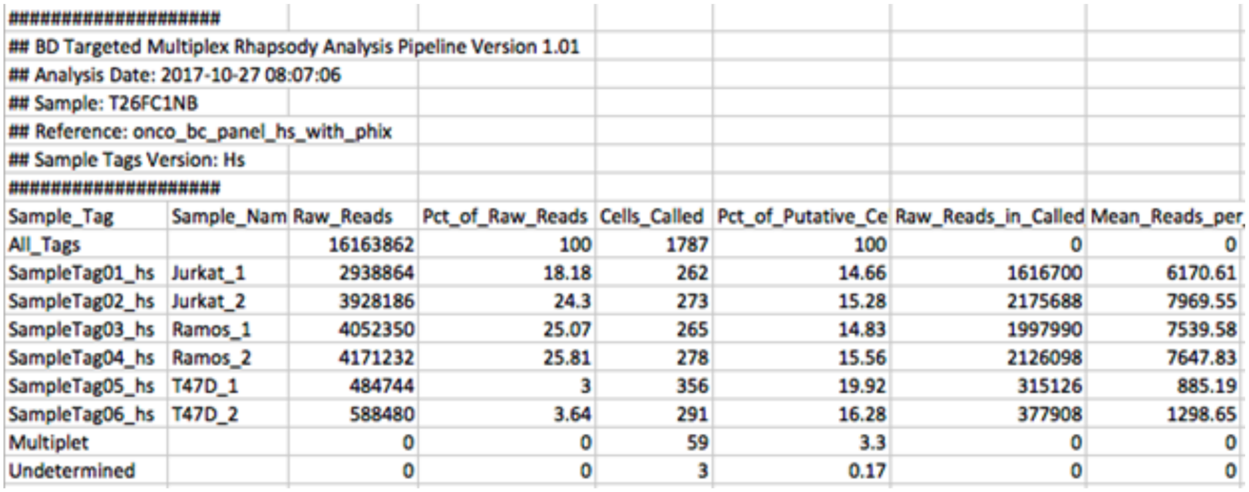
| Metric | Definition |
|---|---|
| Bioproduct | Bioproduct names listed in alphabetical order |
| Raw_Reads | Number of reads before molecular identifier adjustment algorithms |
| Raw_Molecules | Number of UMIs before molecular identifier adjustment algorithms |
| Raw_Seq_Depth | Number of raw reads ÷ the number of raw molecules |
| RSEC_Adjusted_Molecules | Number of molecules detected after RSEC molecular identifier adjustment algorithm |
| RSEC_Adjusted_Reads_non-singleton | Number of RSEC-adjusted reads from molecules represented by more than one read |
| RSEC_Adjusted_Molecules_non-singleton | Number of RSEC-adjusted molecules represented by more than one read |
| RSEC_Adjusted_Seq_Depth | Number of raw reads ÷ the number of RSEC-adjusted molecules |
| DBEC_Adjusted_Reads | Number of reads retained after DBEC molecular identifier adjustment algorithm |
| DBEC_Adjusted_Molecules | Number of molecules retained after RSEC and DBEC |
| DBEC_Minimum_Depth | Threshold of RSEC depth for a molecule to be considered a putative molecule by DBEC |
| DBEC_Adjusted_Seq_Depth | Number of DBEC-adjusted reads ÷ the number of molecules detected after RSEC and DBEC |

| Metric | Definition |
|---|---|
| DBEC_Depth_Status | Bioproduct DBEC correction status:<br>Not detected: Bioproduct was not detected, because it has zero reads<br>Low depth: Minimum sequencing depth not achieved. DBEC not applied.<br>Pass: Minimum sequencing depth has been achieved |

# Sample Tag Metrics

**File**: `[sample_name]_Sample_Tag_Metrics.csv`

The Sample Tag metrics file contains statistics on the reads aligned to each Sample Tag and cells called for each sample. For example:



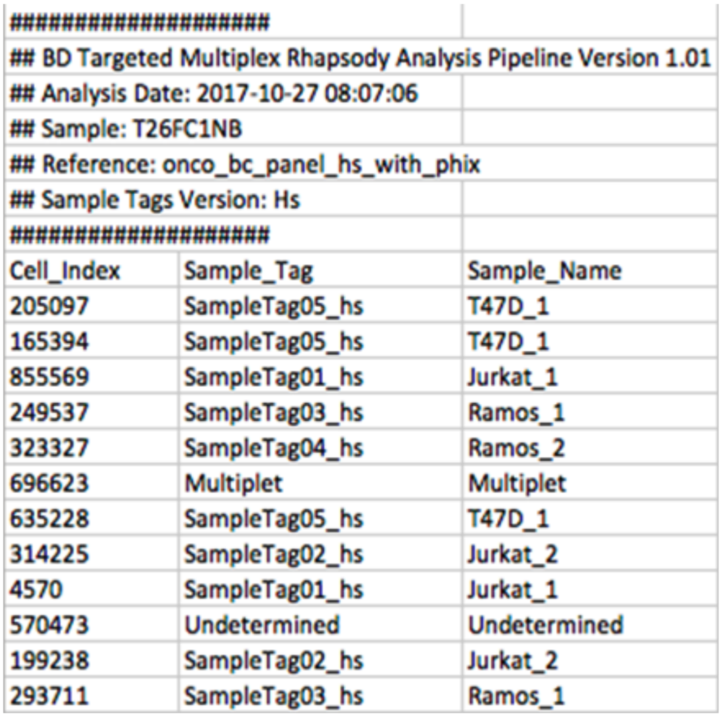
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| #################### | | | | | | | |
| ## BD Targeted Multiplex Rhapsody Analysis Pipeline Version 1.01 | | | | | | | |
| ## Analysis Date: 2017-10-27 08:07:06 | | | | | | | |
| ## Sample: T26FC1NB | | | | | | | |
| ## Reference: onco_bc_panel_hs_with_phix | | | | | | | |
| ## Sample Tags Version: Hs | | | | | | | |
| #################### | | | | | | | |
| Sample_Tag | Sample_Nam | Raw_Reads | Pct_of_Raw_Reads | Cells_Called | Pct_of_Putative_Ce | Raw_Reads_in_Called | Mean_Reads_per_ |
| All_Tags | | 16163862 | 100 | 1787 | 100 | 0 | 0 |
| SampleTag01_hs | Jurkat_1 | 2938864 | 18.18 | 262 | 14.66 | 1616700 | 6170.61 |
| SampleTag02_hs | Jurkat_2 | 3928186 | 24.3 | 273 | 15.28 | 2175688 | 7969.55 |
| SampleTag03_hs | Ramos_1 | 4052350 | 25.07 | 265 | 14.83 | 1997990 | 7539.58 |
| SampleTag04_hs | Ramos_2 | 4171232 | 25.81 | 278 | 15.56 | 2126098 | 7647.83 |
| SampleTag05_hs | T47D_1 | 484744 | 3 | 356 | 19.92 | 315126 | 885.19 |
| SampleTag06_hs | T47D_2 | 588480 | 3.64 | 291 | 16.28 | 377908 | 1298.65 |
| Multiplet | | 0 | 0 | 59 | 3.3 | 0 | 0 |
| Undetermined | | 0 | 0 | 3 | 0.17 | 0 | 0 |

| File | Description | Major contributing factors |
|---|---|---|
| Sample_Tag | List of the Sample Tags in the pipeline run | — |
| Sample_Name | User-provided sample name | — |
| Raw_Reads | Number of reads aligned to each Sample Tag | Sample Tag sequencing amount |
| Pct_of_Raw_Reads | Percentage of Sample Tag reads aligned to each Sample Tag | Sample Tag sequencing amount |
| Cells_Called | Number of putative cells called for each Sample Tag | Number of cells input and captured by cartridge workflow<br>Sample Tag sequencing amount |
| Pct_of_Putative_Cells_Called | Percentage of putative cells called for each Sample Tag | Number of cells input and captured by cartridge workflow<br>Sample Tag sequencing amount |
| Raw_Reads_in_Called_Cells | Number of Sample Tag reads that are assigned to called cells | Sample Tag sequencing amount |
| Mean_Reads_per_Called_Cell | Average number of Sample Tag reads representing each called cell | Sample Tag sequencing amount |

# Sample Tag Calls

**File**: `[sample_name]_Sample_Tag_Calls.csv`

The Sample Tag calls file contains the determined sample call for every putative cell. Sample names that you provided are included in a separate column. The Sample Tag calls file can be used to annotate the main data tables, which contain results from all samples. For example:



| | | |
|---|---|---|
| #################### | | |
| ## BD Targeted Multiplex Rhapsody Analysis Pipeline Version 1.01 | | |
| ## Analysis Date: 2017-10-27 08:07:06 | | |
| ## Sample: T26FC1NB | | |
| ## Reference: onco_bc_panel_hs_with_phix | | |
| ## Sample Tags Version: Hs | | |
| #################### | | |
| Cell_Index | Sample_Tag | Sample_Name |
| 205097 | SampleTag05_hs | T47D_1 |
| 165394 | SampleTag05_hs | T47D_1 |
| 855569 | SampleTag01_hs | Jurkat_1 |
| 249537 | SampleTag03_hs | Ramos_1 |
| 323327 | SampleTag04_hs | Ramos_2 |
| 696623 | Multiplet | Multiplet |
| 635228 | SampleTag05_hs | T47D_1 |
| 314225 | SampleTag02_hs | Jurkat_2 |
| 4570 | SampleTag01_hs | Jurkat_1 |
| 570473 | Undetermined | Undetermined |
| 199238 | SampleTag02_hs | Jurkat_2 |
| 293711 | SampleTag03_hs | Ramos_1 |

| Column | Description |
|---|---|
| Cell_Index | Unique cell identifier |
| Sample_Tag | The sample tag identifed for the cell |
| Sample_Name | User-provided corresponding sample name |

## Sample Tag Folders

**File**: `[sample_name]_Sample_Tag[number].zip`

or

`[sample_name]_Multiplet_and_Undetermined.zip`

Either zipped file includes:

```
[sample_name]_Sample_Tag[number]_RSEC_MolsPerCell_MEX.zip
[sample_name]_Sample_tag[number]_Metric_Summary.csv
```

Each sample with at least one called putative cell will generate a sample-specific folder containing data tables and selected metrics. The formats of the files are the same as described in corresponding output file descriptions.

Data for putative cells that could not be assigned to a specific sample are found in the Multiplet and Undetermined folder.

# VDJ Metrics

**File**: `[sample_name]_VDJ_metrics.csv`

Metrics specific to TCR and BCR data, also broken down by chain and by cell type (experimental).

This file is only output when the experiment included an appropriate TCR/BCR assay, and the `VDJ_Version` option is selected.

## Overall VDJ Metrics

| Metric | Definition | Major contributing factors |
|---|---|---|
| Reads_Cellular_Aligned_to_VDJ | Number of reads with a valid cell label and UMI that aligned to a VDJ gene segment per chain category. | Sequencing quality<br>Library quality |
| Reads_Contig_Assembled | Number of cellular VDJ aligned reads that were assembled into a contig. | Cell viability<br>Library quality |
| Reads_VDJ _Annotated | Number of reads in contigs passing e-value quality filter. | Cell viability<br>Library quality |
| Reads_Putative | Number of Reads_VDJ_Annotated that came from a putative cell. | Cartridge workflow performance |
| Reads_Corrected | Number of putative VDJ reads that are from dominant contigs and remain after distribution-based error correction. | Cell viability<br>Library quality |
| Pct_Reads_Corrected | Percent reads of the preceding metric relative to Reads_Contig_Assembled. | Cell viability<br>Library quality |
| Mean_Reads_Corrected_per_Putative_Cell | Average corrected reads per putative cell. | Cell viability<br>Library quality |
| Molecules_VDJ_Annotated | Number of molecules represented by reads in Reads_VDJ_Annotated metric. | Cell viability<br>Library quality |
| Molecules_Corrected | Number of molecules represented by reads in Reads_Corrected metric. | Cell viability<br>Library quality |
| Mean_Molecules_Corrected_per_Putative_Cell | Average number of molecules per putative cell (Molecules_Corrected_Putative / num putative cells). | Cell viability<br>Library quality |
| Dominant_Contigs_Mean_Nucleotide_Length | Average protein-coding nucleotide length for all dominant contigs. | Library quality |
| Dominant_Contigs_Pct_Full_Length | Percent of dominant contigs from putative cells that are VDJ full length contigs. | Library quality |

| Metric | Definition | Major contributing factors |
|---|---|---|
| Dominant_Contigs_Pct_With_CDR3 | Percent of dominant contigs from putative cells with CDR3. | Library quality |
| Chain_Category | Category for chains such as BCR and TCR. | VDJ recombination |

## Chain type metrics

Chain type metrics are identical to overall metrics except that they are split by VDJ chain type, such as TCR Alpha and BCR Kappa.

## Cell type metrics

| Metric | Definition | Major contributing factors |
|---|---|---|
| Cell_Type_Experimental | Inferred cell type. Cell type is inferred, either from the mRNA targeted panel expression data or from relative counts of BCR vs TCR | Sample type<br>mRNA panel |
| Number_cells | Number of cells classified as this cell type. | Sample type |
| BCR_Paired_Chains_Pct_Any | Percent of cells of each type that had both a BCR heavy chain and BCR light chain (Kappa or Lambda). | Cell viability<br>Library quality |
| TCR_Paired_Chains_Pct_Any | Percent of cells of each type that had either TCR Alpha and TCR Beta, or TCR Gamma and TCR Delta. | Cell viability<br>Library quality |
| BCR_Paired_Chains_Pct_Full | Percent of cells of each type that had full-length contigs for both BCR heavy chain and BCR light chain (Kappa or Lambda). | Cell viability<br>Library quality |
| TCR_Paired_Chains_Pct_Full | Percent of cells of each type that had full-length contigs for either TCR Alpha and TCR Beta, or TCR Gamma and TCR Delta. | Cell viability<br>Library quality |
| [chain_type]_Pct_Cells_Positive | Percent of cells of each cell type that had at least one valid corrected contig of the listed chain type. | Cell viability<br>Library quality |
| [chain_type]_Pct_Cells_Full_Length | Percentage of cells from each cell type which had a full length contig with the listed chain type. | Cell viability<br>Library quality |
| [chain_type]_Mean_Molecules_per_Cell | Mean number of corrected molecules of the listed chain type in each cell type. | Cell viability<br>Library quality |

## High Quality Filtered Cell Metrics

High quality filtered cell metrics are identical to cell type metrics, but only for the subset of cells that were designated high-quality B or T cells, as described in the TCR and BCR Analysis (page 57).

# VDJ Per Cell

**File**: `[sample_name]_VDJ_perCell.csv`

Putative cells only, cell order is the same as gene expression data table (RSEC/DBEC_MolsPerCell_MEX.zip). Only dominant contigs, all error correction applied.

Data columns: Read and molecule counts, VDJ gene segments, CDR3 sequence, pairing, and cell type.

This file is only output when the experiment included an appropriate TCR/BCR assay, and the `VDJ_Version` option is selected.

| **Metric** | **Definition** | **Major contributing factors** |
|---|---|---|
| Cell_Index | Unique cell ID for the cell represented by this row. Cell index will match between VDJ data and gene/AbSeq expression data tables. | Sequencing quality<br>Library quality |
| Total_VDJ_Read_Count | Total number of error-corrected VDJ reads for all chains in the cell. | Cell viability<br>Library quality |
| Total_VDJ_Molecule_Count | Total number of error-corrected VDJ molecules for all chains in the cell. | Cell viability<br>Library quality |
| [chain_type]_V_gene_Dominant | Dominant V gene segment identified for this chain type in the cell. | VDJ recombination |
| [chain_type]_D_gene_Dominant | Dominant D gene segment identified for this chain type in the cell. | VDJ recombination |
| [chain_type]_J_gene_Dominant | Dominant J gene segment identified for this chain type in the cell. | VDJ recombination |
| [chain_type]_C_gene_Dominant | Dominant C gene segment identified for this chain type in the cell. | VDJ recombination |
| [chain_type]_CDR3_Nucleotide_Dominant | Nucleotide sequence of the dominant clone for this chain type in the cell. | VDJ recombination |
| [chain_type]_CDR3_Translation_Dominant | Amino acid sequence of the dominant clone for this chain type in the cell. | VDJ recombination |
| [chain_type]_Read_Count | Number of error-corrected reads for this chain type in the cell. | Cell viability<br>Library quality |
| [chain_type]_Molecule_Count | Number of unique error-corrected molecules (UMI)for this chain type in the cell. | Cell viability<br>Library quality |
| BCR_Paired_Chains | True/False — this cell contains at least one error-corrected molecule of each BCR heavy and light (Kappa or Lambda). | Cell viability<br>Library quality |

| Metric | Definition | Major contributing factors |
|---|---|---|
| TCR_Paired_Chains | True/False — this cell contains at least one error-corrected molecule of each TCR Alpha and TCR Beta, or TCR Gamma and TCR Delta. | Cell viability<br>Library quality |
| Cell_Type_Experimental | Inferred cell type of this cell index. Cell type is inferred, either from the mRNA targeted panel expression data or from relative counts of BCR vs TCR. | Sample type<br>mRNA panel |
| High_Quality_Cell | True/False - this cell was designated as high quality, having a B or T type, a productive contig, and sufficient VDJ molecules | Sample quality<br>Library quality |

# VDJ Per Cell Uncorrected

**File**: `[sample_name]_VDJ_perCell_uncorrected.csv`

All cell IDs – putative and non-putative.

Only dominant contigs, no error correction, no chain family consolidation.

Data columns: Read and molecule counts, VDJ gene segments, CDR3 sequence, pairing, and cell type.

Shared column definitions are identical to the `VDJ_perCell.csv` file. Here are listed the columns unique to this file.

| Metric | Definition | Major contributing factors |
|---|---|---|
| Putative_ Cell | True/False — this cell index was selected as a putative cell based on the mRNA Panel. | Cell viability<br>mRNA Panel |

# VDJ Dominant Contigs AIRR

**File**: `[sample_name]_VDJ_Dominant_Contigs_AIRR.tsv`

Putative cells only, dominant contig for each CellID–chain combination. DBEC adjustment is applied. The file is compliant with the AIRR rearrangement schema and contains additional informational columns in addition to all the mandatory ones.

Data columns: Cell Identifiers, Read and Molecule counts, Full trimmed contig nucleotide and amino acid sequence, Framework and CDR region nucleotide and amino acid sequence, V, D, J, and C gene segments, full length and productive status.

Refer to docs.airr-community.org/en/stable/datarep/rearrangements.html

This file is only output when the experiment included an appropriate TCR/BCR assay, and the `VDJ_Version` option is selected.

| Metric | Definition | Major contributing factors |
|---|---|---|
| cell_id | Unique cell ID for the cell represented by this row. Cell index will match between VDJ data and gene/AbSeq expression data tables | Sequencing quality<br>Library quality |
| cell_type_experimental | Inferred cell type. Cell type is inferred, either from mRNA targeted panel expression data or from relative counts of BCR vs TCR | Sample type<br>mRNA Panel |
| high_quality_cell | True/False - This cell was designated as high quality, having a B or T type, a productive contig, and sufficient VDJ molecules | Sample quality<br>Library quality |
| locus | Type of VDJ sequence: one of TRA, TRB, TRG, TRD, IGH, IGK, and IGL | Cell viability<br>Library quality |
| sequence_id | Unique ID for contig formatted as [cell_id]_[locus]_[number] | Sequencing quality<br>Library quality |
| consensus_count | Number of reads for this contig | Cell viability<br>Library quality |
| umi_count | Number of unique molecules (UMI) for this contig. Previously called "duplicate_count" in an earlier AIRR standard | Cell viability<br>Library quality |
| sequence | Assembled nucleotide sequence of contig after trimming | Library quality<br>VDJ recombination |
| sequence_length | Length of full contig nucleotide sequence after trimming | Library quality<br>VDJ recombination |
| sequence_aa | Amino acid sequence of contig after trimming | Library quality<br>VDJ recombination |

| Metric | Definition | Major contributing factors |
|---|---|---|
| sequence_aa_length | Length of full contig amino acid sequence after trimming | Library quality<br>VDJ recombination |
| sequence_alignment | Nucleotide sequence corresponding to VDJ coding region after trimming | VDJ recombination |
| sequence_alignment_length | Length of nucleotide sequence corresponding to VDJ coding region after trimming | VDJ recombination |
| sequence_alignment_aa | Amino acid sequence corresponding to VDJ coding region after trimming | VDJ recombination |
| sequence_alignment_aa_ length | Length of amino acid sequence corresponding to VDJ coding region after trimming | VDJ recombination |
| junction | Junction region nucleotide sequence, where the junction is defined as the CDR3 plus the two flanking conserved codons | VDJ recombination |
| junction_aa | Amino acid translation of the junction | VDJ recombination |
| productive | True/False — there are no stop codons in the protein-coding portion of the sequence | VDJ recombination |
| rev_comp | True/False — the alignment is on the opposite strand (reverse complemented) with respect to the contig sequence. This field is always False for contig sequences from the BD Rhapsody™ VDJ library | Sequencing quality<br>Library quality |
| complete_VDJ | True/False — this cell chain combination contains some amino acid sequence for each framework (FR1-FR4) region and each CDR (1-3) region | Library quality<br>VDJ recombination |
| v_call | V gene segment identified for this contig | VDJ recombination |
| v_support | Quality of V gene alignment - lower is better | Sequencing quality<br>Library quality |
| v_cigar | CIGAR string for the V gene alignment | VDJ recombination |
| v_sequence_start | Start position of the V gene in the contig sequence (1-based closed interval) | VDJ recombination |
| v_sequence_end | End position of the V gene in the contig sequence (1-based closed interval) | VDJ recombination |
| d_call | First or only D gene segment identified for this contig | VDJ recombination |
| d_support | Quality of D gene alignment, lower is better | Sequencing quality<br>Library quality |
| d_cigar | CIGAR string for the D gene alignment | VDJ recombination |
| d_sequence_start | Start position of the D gene in the contig sequence (1-based closed interval) | VDJ recombination |

| Metric | Definition | Major contributing factors |
|---|---|---|
| d_sequence_end | End position of the D gene in the contig sequence (1-based closed interval) | VDJ recombination |
| j_call | J gene segment identified for this contig | VDJ recombination |
| j_support | Quality of J gene alignment - lower is better | Sequencing quality<br>Library quality |
| j_cigar | CIGAR string for the J gene alignment | VDJ recombination |
| j_sequence_start | Start position of the J gene in the contig sequence (1-based closed interval) | VDJ recombination |
| j_sequence_end | End position of the J gene in the contig sequence (1-based closed interval) | VDJ recombination |
| c_call | C gene segment identified for this contig | VDJ recombination |
| fwr1 | Nucleotide sequence of the FR1 for the contig | VDJ recombination |
| fwr1_aa | Amino acid sequence of the FR1 for the contig | VDJ recombination |
| fwr2 | Nucleotide sequence of the FR2 for the contig | VDJ recombination |
| fwr2_aa | Amino acid sequence of the FR2 for the contig | VDJ recombination |
| fwr3 | Nucleotide sequence of the FR3 for the contig | VDJ recombination |
| fwr3_aa | Amino acid sequence of the FR3 for the contig | VDJ recombination |
| fwr4 | Nucleotide sequence of the FR4 for the contig | VDJ recombination |
| fwr4_aa | Amino acid sequence of the FR4 for the contig | VDJ recombination |
| cdr1 | Nucleotide sequence of the CDR1 for the contig | VDJ recombination |
| cdr1_aa | Amino acid sequence of the CDR1 for the contig | VDJ recombination |
| cdr2 | Nucleotide sequence of the CDR2 for the contig | VDJ recombination |
| cdr2_aa | Amino acid sequence of the CDR2 for the contig | VDJ recombination |
| cdr3 | Nucleotide sequence of the CDR3 for the contig | VDJ recombination |
| cdr3_aa | Amino acid sequence of the CDR3 for the contig | VDJ recombination |
| germline_alignment | Assembled, aligned, full-length inferred germline sequence spanning the same region as the sequence_alignment field | VDJ recombination |
| germline_alignment_aa | Amino acid translation of the assembled germline sequence | VDJ recombination |
| v_germline_alignment | Aligned V gene germline sequence spanning the same region as the v_sequence_alignment field and including the same set of corrections and spacers (if any) | VDJ recombination |
| v_germline_alignment_aa | Amino acid translation of the v_germline_alignment field | VDJ recombination |

| Metric | Definition | Major contributing factors |
|---|---|---|
| d_germline_alignment | Aligned D gene germline sequence spanning the same region as the d_sequence_alignment field and including the same set of corrections and spacers (if any) | VDJ recombination |
| d_germline_alignment_aa | Amino acid translation of the d_germline_alignment field | VDJ recombination |
| j_germline_alignment | Aligned J gene germline sequence spanning the same region as the j_sequence_alignment field and including the same set of corrections and spacers (if any) | VDJ recombination |
| j_germline_alignment_aa | Amino acid translation of the j_germline_alignment field | VDJ recombination |
| v_germline_start | Alignment start position in the V gene reference sequence (1-based closed interval) | VDJ recombination |
| v_germline_end | Alignment end position in the V gene reference sequence (1-based closed interval) | VDJ recombination |
| d_germline_start | Alignment start position in the D gene reference sequence for the first or only D gene (1-based closed interval) | VDJ recombination |
| d_germline_end | Alignment end position in the D gene reference sequence for the first or only D gene (1-based closed interval) | VDJ recombination |
| j_germline_start | Alignment start position in the J gene reference sequence (1-based closed interval) | VDJ recombination |
| j_germline_end | Alignment end position in the J gene reference sequence (1-based closed interval) | VDJ recombination |
| np1_length | Nucleotide sequence length of the combined N/P region between the V gene and first D gene alignment or between the V gene and J gene alignments | VDJ recombination |
| np2_length | Nucleotide sequence length of the combined N/P region between either the first D gene and J gene alignments or the first D gene and second D gene alignments | VDJ recombination |

# VDJ Unfiltered Contigs AIRR

**File**: `[sample_name]_VDJ_Unfiltered_Contigs_AIRR.tsv`

All cell IDs, all assembled contigs that were successfully annotated.

The file is compliant with the AIRR rearrangement schema and contains additional informational columns in addition to all the mandatory ones.

Data columns: Cell Identifiers, Read and Molecule counts, Full trimmed contig nucleotide and amino acid sequence, Framework and CDR region nucleotide and amino acid sequence, V, D, J, and C gene segments, full length, and productive status.

Refer to docs.airr-community.org/en/stable/datarep/rearrangements.html

Shared column definitions are identical to the VDJ_Dominant_Contigs_AIRR.tsv file. Here are listed the columns unique to this file.

| Metric | Definition | Major contributing factors |
|---|---|---|
| Dominant | True/False — this contig was selected as the dominant contig for this cell-chain combination. | Library quality<br>VDJ recombination |
| Putative_ Cell | True/False — this cell index was selected as a putative cell based on the mRNA panel. | Cell viability<br>mRNA panel |

# Protein Aggregates Experimental

**File**: `[sample_name]_Protein_Aggregates_Experimental.csv`

Cell labels suspected of resulting from protein aggregates are set to TRUE.

This output file only exists when the pipeline parameter `Putative_Cell_Call` is set to AbSeq.

| Column | Description |
|---|---|
| Cell_Index | Unique cell identifier |
| ProteinAggregate | TRUE/FALSE |

# Immune Cell Classification Result

**File**: `[sample_name]_(assay)_cell_type_experimental.csv`

When immune cell classification is done

- **using WTA data**: `[sample_name]_cell_type_experimental.csv`
- **using ATAC data**: `[sample_name]_ATAC_cell_type_experimental.csv`
- **using WTA and ATAC data**: `[sample_name]_Joint_cell_type_experimental.csv`

Immune cell classification result has two colums: cell index and predicted cell type. Here is the list of predicted cell types: classical monocyte, nonclassical monocyte, dendritic cell, B cell, naive CD4 T cell, memory CD4 T cell, naive CD8 T cell, memory CD8 T cell, natural killer cell, and gamma-delta T cell.

# Dimensionality Reduction Coordinates

**File**: `[sample_name]_(method)_(tSNE/UMAP)_coordinates.csv`

When dimensionality reduction is done

- **using WTA data**: `[sample_name]_(tSNE/UMAP)_coordinates.csv`
- **using ATAC data**: `[sample_name]_ATAC_(tSNE/UMAP)_coordinates.csv`
- **using WTA and ATAC data**: `[sample_name]_Joint_(tSNE/UMAP)_coordinates.csv`

The csv file has three columns: cell index, two tSNE or UMAP coordinates.

# ATAC Fragments and Fragments Index

**Files**:

```
[sample_name]_ATAC_Fragments.bed.gz
[sample_name]_ATAC_Fragments.bed.gz.tbi
```

The Fragments file is a BED3+2 tabular file that has one line for each fragment, sorted by chromosome and position. The first three columns are the same that all BED files use, making this file function as a BED file for many purposes, while the last two diverge from the BED specification. The fourth column is the fragment's cell index, identifying the specific BD Rhapsody™ bead that bound the fragment. The fifth column shows the total number of aligned read-pairs associated with the fragment, including any duplicates.

| Column | Name | Description |
|---|---|---|
| 1 | chrom | Chromosome or scaffold identifier |
| 2 | chromStart | Zero-indexed Tn5-adjusted start site of fragment |
| 3 | chromEnd | Zero-indexed Tn5-adjusted end site of fragment (non-inclusive) |
| 4 | cellIndex | BD Rhapsody™ bead index (integer) |
| 5 | readSupport | Count of read-pairs that support the existence of this fragment |

The Fragments Index is the index file generated by tabix for the coordinate-sorted fragments file.

# ATAC Transposase Sites and Index file

**Files**:

```
[sample_name]_ATAC_Transposase_Sites.bed.gz
[sample_name]_ATAC_Transposase_Sites.bed.gz.tbi
```

The Transposase Sites file is a BED3+2 tabular file that has one line per transposase cut site (represented by a base at each end of each fragment), sorted by chromosome and position. The first three columns are the same that all BED files use, making this file function as a BED file for many purposes, while the last two diverge from the BED specification. The fourth column is the fragment's cell barcode, identifying the specific BD Rhapsody™ bead that bound the fragment. The fifth column shows the total number of aligned read-pairs associated with the fragment, including any duplicates.

| Column | Name | Description |
|---|---|---|
| 1 | chrom | Chromosome or scaffold identifier |
| 2 | chromStart | Zero-indexed location of transposase site |
| 3 | chromEnd | chromStart + 1 |
| 4 | cellIndex | BD Rhapsody™ bead index (integer) |
| 5 | readSupport | Count of read-pairs that support the existence of the associated fragment |

The Transposase Sites Index is the index file generated by tabix for the coordinate-sorted Transposase Sites file.

# ATAC Peaks and Peaks Index

**Files**:

```
[sample_name]_ATAC_Peaks.bed.gz
[sample_name]_ATAC_Peaks.bed.gz.tbi
```

The Peaks file is the bgzipped narrowPeak output from MACS2 detecting peak regions of Tn5 transposase activity. The file is a BED6+4 tabular file that has one line for each peak, sorted by chromosome and position. The columns are

| Column | Name | Description |
|---|---|---|
| 1 | chrom | Chromosome or scaffold identifier |
| 2 | chromStart | Zero-indexed start site of peak region |
| 3 | chromEnd | Zero-indexed end site of peak region (non-inclusive) |
| 4 | name | Name of the peak |
| 5 | score | Integer score for display, calculated as int(-10*log10qvalue) |
| 6 | strand | Always a "." because these peak regions are not strand-specific |
| 7 | - | Fold-change at peak summit |
| 8 | - | -log10pvalue at peak summit |
| 9 | - | -log10qvalue at peak summit |
| 10 | - | Summit position relative to peak start |

The Fragments Index is the index file generated by tabix for the coordinate-sorted Peaks file.

# ATAC Cell-by-Peak Data Tables

Files containing putative cells only:

```
[sample_name]_ATAC_Cell_by_Peak_MEX.zip
```

Unfiltered file containing all cell indexes with >=1 transposase sites in peaks:

```
[sample_name]_ATAC_Cell_by_Peak_Unfiltered_MEX.zip
```

The number of transposase sites from each cell that fall within each peak region is represented in the matrix market exchange (MEX) format. The MEX format is an efficient way to store sparse data, and is a common input format for many single-cell analysis tools. The MEX.zip output files contain three separate gzip compressed files that together represent the number of transposase sites of each peak in each cell. By convention, these files are named:

- **atac-barcodes.tsv**: Containing a list of cell indexes (integer between 1 and 3843), one per row.
- **atac-features.tsv**: Containing a list of peaks detected, one per row. For improved compatibility, this file contains three columns, the first two of which are a duplicated peak coordinate that follows a format of `[chromosome]:[start]-[end]`, and a third which indicates the features are peaks.
- **atac-matrix.mtx**: Containing a three column per row representation of the number of transposase sites in peaks. First column is the 1-based row number from atac-features.tsv (peak). Second column is the 1-based row number from atac-barcodes.tsv (cell). Third column is the number of transposase sites detected for that peak in that cell.

Cell indexes in the atac-barcodes.tsv file are sorted numerically.

# ATAC Metrics

**Files**:

```
[sample_name]_ATAC_Metrics.json
[sample_name]_ATAC_Metrics.csv
```

## Metric definitions

### Sequencing Quality

| Metric | Definition | Majo contributing factors |
|---|---|---|
| Total_Reads_in_FASTQ | Number of (R1, R2, I2) reads in the ATAC input FASTQ files | Sequencing amount |
| Reads_Too_Short | Number of reads filtered out due to length of either I2 (minimum to derive cell label) or R1/R2 (<30 bp) | Sequencing quality |
| Pct_Reads_Too_Short | Percentage of reads filtered out due to length of either I2 (minimum to derive cell label) or R1/R2 (<30 bp) | Sequencing quality |
| Reads_Low_Base_Quality | Number of reads filtered out due to average base quality score <20 | Sequencing quality |
| Pct_Reads_Low_Base_Quality | Percentage of reads filtered out due to average base quality score <20 | Sequencing quality |
| Reads_High_SNF | Number of reads filtered out due to single nucleotide frequency >=55% for I2 or >=80% for R1 or R2 - low complexity reads will not result in a useful cell label or alignment | Sequencing quality |
| Pct_Reads_High_SNF | Percentage of reads filtered out due to single nucleotide frequency >=55% for I2 or >=80% for R1 or R2 - low complexity reads will not result in a useful cell label or alignment | Sequencing quality |
| Pct_Reads_Filtered_Out | Percentage of reads removed by the combination of length, quality, and SNF filters | Sequencing quality |
| Library | Name of library | Name of library |

## Library Quality

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_Filtered_Reads | Number of reads after length, quality, and SNF filtering | Sequencing amount<br>Sequencing quality<br>Library quality |
| Pct_Q30_Bases_in_Filtered_R1 | Percentage of R1 bases with quality score >30, averaged across all reads retained after quality filtering | Sequencing quality |
| Pct_Q30_Bases_in_Filtered_R2 | Percentage of R2 bases with quality score >30, averaged across all reads retained after quality filtering | Sequencing quality |
| Pct_CellLabel | Percentage of filtered reads with a valid cell label (I2) | Sequencing quality<br>Library quality |
| Reads_CellLabel_Aligned_Confidently | Number of filtered reads containing a valid cell label (I2) that aligned with Mapping Quality of R1 and R2 >= 30 | Sequencing quality<br>Library quality |
| Pct_CellLabel_Aligned_Confidently | Percentage of filtered reads containing a valid cell label (I2) that aligned with Mapping Quality of R1 and R2 >= 30 | Sequencing quality<br>Library quality |
| Pct_Useful_ATAC_Reads | Percentage of filtered reads containing a valid cell label (I2) that aligned confidently (Mapping Quality of R1 and R2 >= 30) to a non-mitochondrial chromosome with an appropriate fragment size (>= 10 and <= 5000) | Sequencing quality<br>Library quality<br>Sample preparation |
| Library | Name of library | Name of library |

## Alignment Categories

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_CellLabel_Reads | Number of filtered reads with a valid cell label (I2) | Sequencing quality<br>Library quality |
| Total_Nuclear_Fragment_Reads | Number of cell label reads that aligned confidently (MQ of R1 and R2 >= 30) to a non-mitochondrial chromosome with an appropriate fragment size (>= 10 and <= 5000) | Sequencing quality<br>Library quality<br>Sample preparation |
| Pct_Nuclear_Fragments | Percentage of cell label reads that aligned confidently (Mapping Quality of R1 and R2 >= 30) to a non-mitochondrial chromosome with an appropriate fragment size (>= 10 and <= 5000) | Sequencing quality<br>Library quality<br>Sample preparation |

| Metric | Definition | Major contributing factors |
|---|---|---|
| Reads_Invalid_Fragments | Number of cell label reads that aligned confidently (Mapping Quality of R1 and R2 >= 30) but R1 and R2 aligned to different chromosomes or had an inappropriate fragment size (<= 10 or >= 5000) | Sequencing quality<br>Library quality |
| Pct_Invalid_Fragments | Percentage of cell label reads that aligned confidently (Mapping Quality of R1 and R2 >= 30) but R1 and R2 aligned to different chromosomes or had an inappropriate fragment size (<= 10 or >= 5000) | Sequencing quality<br>Library quality |
| Reads_Mitochondrial | Number of cell label reads that aligned confidently (Mapping Quality of R1 and R2 >= 30) to the mitochondrial chromosome | Sequencing quality<br>Library quality<br>Sample preparation |
| Pct_Mitochondrial | Percentage of cell label reads that aligned confidently (Mapping Quality of R1 and R2 >= 30) to the mitochondrial chromosome | Sequencing quality<br>Library quality<br>Sample preparation |
| Reads_Aligned_Not_Confidently | Number of cell label reads that did not align confidently (Mapping Quality of R1 or R2 < 30) to reference | Sequencing quality<br>Library quality |
| Pct_Aligned_Not_Confidently | Percentage of cell label reads that did not align confidently (Mapping Quality of R1 or R2 < 30) to reference | Sequencing quality<br>Library quality |
| Reads_Unaligned | Number of cell label reads with either R1 or R2 unaligned to reference | Sequencing quality<br>Library quality |
| Pct_Unaligned | Percentage of cell label reads with either R1 or R2 unaligned to reference | Sequencing quality<br>Library quality |
| Library | Name of library | Name of library |

## Fragments

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_Nuclear_Fragment_Reads | Number of cell label reads that aligned confidently (MQ of R1 and R2 >= 30) to a non-mitochondrial chromosome with an appropriate fragment size (>= 10 and <= 5000) | Sequencing quality<br>Library quality<br>Sample preparation |
| Total_Nonduplicate_Fragments | Number of fragments remaining after removing duplicate fragments and cell labels that have zero fragments in peak regions | Sequencing quality<br>Library quality |

| Metric | Definition | Major contributing factors |
|---|---|---|
| Pct_Duplicate_Fragments | Percentage of total nuclear fragment reads that were found to be duplicates | Sequencing quality<br>Library quality |
| Nonduplicate_Fragments_from_Cell_Labels | Number of non-duplicate fragments associated with cell labels that have at least one fragment that overlaps a peak region | Sequencing quality<br>Library quality |
| Pct_Nonduplicate_Fragments_from_Cell_Labels | Percentage of non-duplicate fragments associated with cell labels that have at least one fragment that overlaps a peak region | Sequencing quality<br>Library quality |
| Pct_Nonduplicate_Fragments_with_NFR_Lengths | Percentage of non-duplicate fragments with nucleosome-free-region (NFR) lengths < 147 bp | Library quality<br>Sample preparation |
| Pct_Nonduplicate_Fragments_with_Mononucleosomal_Lengths | Percentage of non-duplicate fragments with mononucleosome lengths (>= 147 bp and <= 294 bp) | Library quality<br>Sample preparation |

## Peaks

| Metric | Definition | Major contributing factors |
|---|---|---|
| Total_Transposase_Sites | Number of transposase sites | Total_Nonduplicate_Fragments |
| Total_Peaks | Number of peaks (genomic regions of Tn5-accessible chromatin) identified | Library quality<br>Sample preparation<br>Sequencing depth |
| Total_Peak_Basepairs | Total basepair width of all the called peaks combined | Genome size<br>Cell Type<br>Library quality |
| Pct_Genome_Within_Peaks | Percentage of total reference genome width that falls within chromatin accessibility peak regions | Cell Type<br>Library quality |
| Fraction_of_Fragments_Overlapping_Peaks | Percentage of non-duplicate fragments overlapping peak regions | Library quality<br>Sample preparation<br>Pct_Genome_Within_Peaks |
| Fraction_of_Transposase_Sites_in_Peaks | Percentage of transposase sites that fall within peak regions | Library quality<br>Sample preparation<br>Pct_Genome_Within_Peaks |

| Metric | Definition | Major contributing factors |
|---|---|---|
| TSS_Enrichment_Score | Transcription Start Site (TSS) enrichment score: calculated from aggregate distribution of transposase sites in the region extending 2000bp in either direction of all annotated TSSs, normalized against the (background) average number of transposase sites in the first and last 100bp of the 4001bp region | Library quality<br>Sample preparation<br>Cell type |

## Putative Cells

| Metric | Definition | Majo contributing factors |
|---|---|---|
| Total_Putative_Cells | Total number of cell indexes that were identified as likely corresponding to individual cell nuclei in the sample | Library quality<br>Sample preparation |
| Pct_Reads_from_Putative_Cells | Percentage of ATAC reads assigned to putative cells | Library quality |
| Mean_Reads_per_Cell | Average number of ATAC reads per putative cell | Library quality<br>Sequencing depth |
| Median_Reads_per_Cell | Median number of ATAC reads per putative cell | Library quality<br>Sequencing depth |
| Median_Pct_Nonredundant_Read_Pairs_per_Cell | Median value of the percentage of ATAC reads from each putative cell that non-redundantly identify an ATAC fragment | Library quality<br>Sequencing depth |
| Median_Nonduplicate_Fragments_per_Cell | Median number of non-duplicate fragments per cell | Library quality<br>Sequencing depth |
| Total_Nonduplicate_Fragments_from_Putative_Cells | Number of non-duplicate fragments associated with putative cells | Library quality<br>Sequencing depth |
| Pct_Nonduplicate_Fragments_from_Putative_Cells | Percentage of non-duplicate fragements assigned to putative cells | Library quality<br>Sequencing depth |
| Pct_Cellular_Fragments_Overlapping_Peaks | Percentage of non-duplicate fragments from putative cells that overlap peak regions | Library quality |
| Pct_Cellular_Transposase_Sites_in_Peaks | Percentage of transposase sites from putative cells that fall within peak regions | Library quality |

# 5

# Troubleshooting Guide

This section describes how to troubleshoot pipeline runs that have did not successfully complete, or successful runs where the results were not as expected.

- Access Log Files (page 112)
- Output Metrics and Associated Problems (page 114)
- Skim Sequencing for Assessing Library Quality (page 117)

# Access Log Files

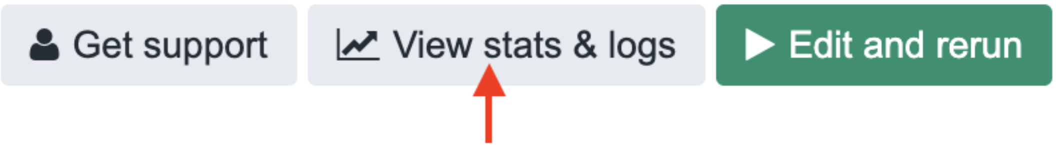
When a Seven Bridges task or local pipeline run fails, it is important to get the log files to help determine the root cause.

## Arranging BD Biosciences to join the project on Seven Bridges Genomics

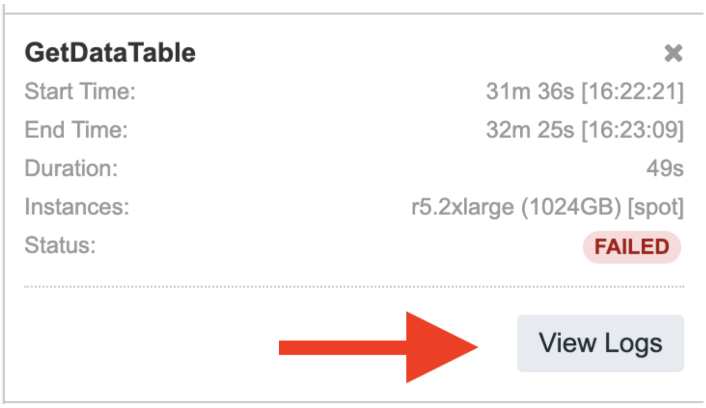
If a task fails on the Seven Bridges Genomics platform, contact BD Biosciences technical support at scomix@bdscomix.bd.com to troubleshoot the issue. Tech support will provide you with instructions on inviting a support team member to your project. To troubleshoot the issue yourself, access the log files.

## Downloading the log file from Seven Bridges Genomics

1. When viewing a failed task, click **View Stats & Logs** in the upper right corner:



2. Locate the failed node in your pipeline run. Completed nodes are in green, and the failed node is in red. Click the failed node, and on the right, click **View Logs** for that node:

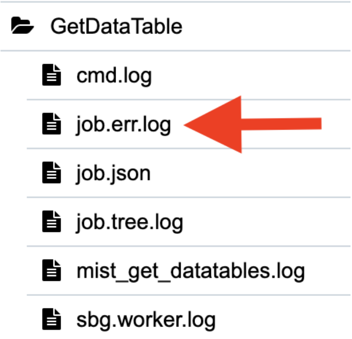


GetDataTable

| | |
|---|---|
| Start Time: | 31m 36s [16:22:21] |
| End Time: | 32m 25s [16:23:09] |
| Duration: | 49s |
| Instances: | r5.2xlarge (1024GB) [spot] |
| Status: | FAILED |

A list of files contained in the failed node are displayed.

3. Click **job.err.log** to display the log content and download it:



GetDataTable

- cmd.log
- job.err.log
- job.json
- job.tree.log
- mist_get_datatables.log
- sbg.worker.log

## Accessing the log file in a local installation

If a pipeline run completed successfully, all logs are collected in a Logs folder in your output directory. But if a pipeline run fails, the Logs folder is absent from the directory. You need to navigate to the *tmp* directory containing the intermediate files for that node to obtain the log files:

1. In the terminal STDOUT, find the failed node command call from CWL-runner. This is the most recent command call.

2. Locate the tmp folder name, which is in the format:

   ```
   [job Name_of_failed_node] /tmp/tmpb0kyIg $
   ```

3. Navigate to that directory. The log file will have the `.log` extension.

4. Send the log file to scomix@bdscomix.bd.com, or contact BD Biosciences technical support without it.

# Output Metrics and Associated Problems

This topic describes possible problems and recommended solutions for sequencing analysis issues. Issues with sequencing metrics might be related to issues that can be resolved in the experimental workflow.

## Percentage reads with cell label (Pct_CellLabel-UMI) and percentage aligned uniquely (Pct_CellLabel_UMI_Aligned_Uniquely) are both low

| Possible causes | Recommended solutions |
|---|---|
| Low sequencing quality | • Ensure that the appropriate PhiX % is used for the type of sequencer used.<br>• Ensure that the Illumina sequencing flow cell is not over-clustered.<br>• Repeat the sequencing run if sequencing quality is suspected to be the reason. |
| Low library quality | • Ensure that the correct panel is used to amplify the sample and the correct amplification protocol and PCR product purification protocols are used.<br>• Repeat amplification from leftover PCR1 products, if necessary. |

## High percentage reads with cell label (Pct_CellLabel-UMI) but low percentage aligned uniquely (Pct_CellLabel_UMI_Aligned_Uniquely)

| Possible causes | Recommended solutions |
|---|---|
| Incorrect FASTA file panel used for mapping | • If <50% alignment, then the wrong panel was likely used.<br>• Verify that the correct panel reference file was used. |
| Incorrect number of sequencing cycles | • Run at least 75 x 2 sequencing cycles. The total length of both reads must be at least 102 bp.<br>• Repeat amplification from leftover PCR1 products, if necessary. |
| Low sequencing quality | Rerun sequencing, and use at least the minimum recommended concentration of PhiX. |

## Low percentage reads from putative cells

| Possible causes | Recommended solutions |
|---|---|
| Some cells in the samples are not well represented by the panel. Their associated cell labels have very few detectable molecules, so they are classified as noise cell labels | • Ensure that the panel matches the sample and species.<br>• Ensure that the panel of genes provides good representation across the cells in the sample tested if all cells are to be detected. |
| Lysis time too long | Ensure that lysis time is exactly 2 minutes and lysis buffer is cold. |

| Possible causes | Recommended solutions |
|---|---|
| Automated pipette settings are incorrect | Ensure that the correct setting is used for the specific step in the cartridge workflow. |
| Wrong buffer used for bead retrieval from the cartridge | Use only lysis buffer, as indicated in the protocol for bead retrieval. |
| Mixed species in experiment | Ensure that the panel used contains genes that cover both species. |
| Excessive dead or dying cells | Proceed with the experiment if cell viability is ≥50%. |
| Very low bead loading density. The bead loading efficiency on the BD Rhapsody$^{TM}$ Scanner likely reported failed. | See bead loading density troubleshooting in the BD Rhapsody$^{TM}$ Single-Cell Analysis System Instrument User Guide (23-21336) or the BD Rhapsody$^{TM}$ Express Single-Cell Analysis System Instrument User Guide (23-21332). |

## Number of cells detected in sequencing is much lower or higher than the expected cell number based on imaging results

| Possible causes | Recommended solutions |
|---|---|
| Wrong inflection point chosen on cell calling graph | Some cell samples or noise profiles create multiple inflection points on the cell calling second-derivitive curve. Guide the basic cell calling algorithm to choose the correct inflection point by running the pipeline with the `Expected_Cell_Count` parameter. Usually this can be the number of cells loaded into the BD Rhapsody™ cartridge. |
| For Targeted mRNA assays: Some cells in the samples are not well represented by the panel. Their associated cell labels have very few detectable molecules, so they are classified as noise cell labels. | • If all of the cells are to be detected, ensure that the panel of genes provides good representation across the cells in the sample tested.<br>• Ensure that the panel matches the sample and species.<br>• If there is more than one bioproduct type in libraries, use another Putative Cell Calling option to troubleshoot. |
| Cell Capture Beads settled to the bottom of the tube before the start of PCR1. | Ensure that Cell Capture Beads are well suspended just before starting PCR1, and the thermal cycler lid is pre-heated when the PCR tubes are placed on the thermal cycler. |
| Cell Capture Beads are lost during handling after cartridge use. | Ensure maximum recovery of Cell Capture Beads by using low retention tips and tubes. See product information in the BD Rhapsody$^{TM}$ Single-Cell Analysis System Instrument User Guide (23-21336) or the BD Rhapsody$^{TM}$ Express Single-Cell Analysis System Instrument User Guide (23-21332). |

## Batch effects across multiple libraries

| Possible causes | Recommended solutions |
|---|---|
| Variations in sequencing depth | For Targeted mRNA assays: Examine the status of each bioproduct in [sample_name]_Bioproduct_Stats.csv across samples. If there are highly abundant genes with a pass status in one library but a low depth status in another, consider using [sample_name]_RSEC_MolsPerCell.csv for analysis. Or, use [sample_name]_DBEC_MolsPerCell.csv for analysis after removal of genes that do not have pass status in any of the libraries under consideration. |
| Variations in cell sample handling protocol | Use a similar cell sample handling protocol for all samples to be analyzed together, noting that temperature, duration of handling, and handling method can affect bioproduct expression. |
| Differences in thermal cycling | For samples to be analyzed together, it is recommended to perform the PCR amplification of the Cell Capture Beads of those samples in parallel. |
| Low sequencing depth | For Targeted mRNA assays: Use [sample_name]_RSEC_MolsPerCell.csv or use [sample_name]_DBEC_MolsPerCell.csv after removal of genes that do not have *pass* status. |

# Skim Sequencing for Assessing Library Quality

Before committing to a large expensive sequencing run, some may prefer to get a preview of the quality of the results by sequencing at low depth. Several output metrics from the BD Rhapsody™ Sequence Analysis Pipeline can be evaluated while performing skim sequencing to assess library and sequencing run quality. Output metrics are stable at low sequencing depth (~ 2 million sequencing reads or higher).

## Metrics that are often stable with skim sequencing

### Sequence Quality

- Pct_Read_Pair_Overlap
- Pct_Reads_Too_Short
- Pct_Reads_Low_Base_Quality
- Pct_Reads_High_SNF
- Pct_Reads_Filtered_Out

### Library Quality

- Pct_Q30_Bases_in_Filtered_R2
- Pct_CellLabel_UMI
- Pct_CellLabel_UMI_Aligned_Uniquely

### Cells

- Putative_Cell_Count
- Pct_Reads_from_Putative_Cells

# 6

# Additional Resources

Useful resources related to the BD Rhapsody™ Sequence Analysis Pipeline and single-cell analysis:

- Extra Utilities (page 120)
- Scripts for BD Rhapsody™ Data (page 124)
- Pipeline Install Bundle (Docker-Free) (page 125)

# Extra Utilities

The local version of the BD Rhapsody™ Sequence Analysis Pipeline comes with several useful utilities:

- Make BD Rhapsody™ Reference (page 120)
- PhiX contamination detection (page 121)
- Annotate Cell Label and UMI only (page 122)

These utilities can be run in the same way as the main Sequence Analysis Pipeline, using cwl-runner and docker. They use the same docker image as the main Sequence Analysis Pipeline -- `bdgenomics/rhapsody`. See Local Server Setup (page 20) for installation instructions. Inputs are provided in a YML specification file, or on the command-line. CWL documents for these utilities are also in the same location as the main pipeline CWL (versioned folders): .

## Make BD Rhapsody™ Reference

### make_rhap_reference_[version].cwl

Create a new WTA Reference Archive for use as an input to the BD Rhapsody™ Sequence Analysis Pipeline.

### Inputs:

- **Genome_fasta**:

  Required. File path to the reference genome file in FASTA or FASTA.GZ format.

- **Gtf**:

  Required. File path to the transcript annotation files in GTF or GTF.GZ format. The Sequence Analysis Pipeline requires the “gene_name” or “gene_id” attribute to be set on each gene and exon feature. Gene and exon feature lines must have the same attribute, and exons must have a corresponding gene with the same value. For TCR/BCR assays, the TCR or BCR gene segments must have the “gene_type” or “gene_biotype” attribute set, and the value should begin with “TR” or “IG”, respectively.

- **Extra_sequences**:

  Optional. File path to additional sequences in FASTA format to use when building the STAR index. (e.g. transgenes or CRISPR guide barcodes). GTF lines for these sequences will be automatically generated and combined with the main GTF.

- **Filtering_off**:

  Optional. [True/False] By default the input GTF files are filtered based on the gene_type/gene_biotype attribute. (Using biotypes defined by Gencode/Ensembl) If you have already pre-filtered the input Annotation files or wish to turn-off the filtering, set this option to True. The GTF features having the following attribute values are are kept:

  > protein_coding, lncRNA (lincRNA and antisense for Gencode < v31/M22/Ensembl97), IG_LV_gene, IG_V_gene, IG_V_pseudogene, IG_D_gene, IG_J_gene, IG_J_pseudogene, IG_

C_gene, IG_C_pseudogene, TR_V_gene, TR_V_pseudogene, TR_D_gene, TR_J_gene, TR_J_pseudogene, TR_C_gene

- **Archive_prefix**:

  Optional. String. A prefix base name for the result compressed archive file. The default value is constructed based on the input reference files.

- **Maximum_threads**:

  Optional. Integer. The maximum number of threads to use. By default, all available cores are used.

- **Extra_STAR_params**

  Optional. String. Parameters to pass directly to the STAR genomeGenerate process. Useful for very large or very small genome sizes. Example "--limitGenomeGenerateRAM 48000 --genomeSAindexNbases 11"

### Example command:

```
cwl-runner make_rhap_reference_2.0.cwl --Genome_fasta GRCh38.primary_
assembly.genome.fa.gz --Gtf gencode.v42.primary_assembly.annotation.gtf.gz
```

### File structure of the resulting reference archive:

```
BD_Rhapsody_Reference_Files/
   star_index/
       [files created with star genomeGenerate]
   [filtered/non-filtered transcriptome annotation].gtf
```

## PhiX contamination detection

### PhiXContamination_[version].cwl

Check a FASTQ file for PhiX contamination, by aligning the reads to the PhiX genome. (uses Bowtie2)

### Inputs:

- **Fastq**:

  Required. File path to a single FASTQ file to check for PhiX contamination.

- **Threads**:

  Optional. Integer. The number of threads to use. By default, all available cores are used.

### Example command:

```
cwl-runner PhiXContamination_2.0.cwl --Fastq MyRhapsodyLibrary_R1.fastq.gz
--Threads 8
```

### Example result:

```
36508493 reads; of these:
  36508493 (100.00%) were unpaired; of these:
    36503405 (99.99%) aligned 0 times
    5088 (0.01%) aligned exactly 1 time
    0 (0.00%) aligned >1 times
0.01% overall alignment rate
```

## Annotate Cell Label and UMI only

### AnnotateCellLabelUMI_[version].cwl

Given pairs of R1/R2 FASTQ files from BD Rhapsody™ libraries, only annotate the cell label and UMI of R1 and put it in the header of R2.

Format of result FASTQ:

```
@OriginalHeader;cell_index;UMI
[R2Sequence]
+
[R2Quality]
```

### Inputs:

- **Reads**:

  Required. Comma-separated list of FASTQ file paths.

- **Maximum_Threads**:

  Optional. Integer. The maximum number of threads to use. By default, all available cores are used.

### Example YML input specification file [inputs.yml]:

```
Reads:
 - class: File
   location: "test/mySample_R1_.fastq.gz"
 - class: File
   location: "test/mySample_R2_.fastq.gz"
Maximum_Threads: 8
```

### Example command:

```
cwl-runner AnnotateCellLabelUMI_2.0.cwl inputs.yml
```

### Example result in mySample_R2.annotated.fastq.gz:

```
@M04277:241:000000000-B4VBL:1:1101:9821:2660;8144695;ATGCACGC
TGCCCTCAACGACCACTTTGTCAAGCTCATTTCCTGGTATGACAACGAATTTGGCTACAGCAACAGGGTGGTGGAC
+
CCCCCGGFFGGGGGDGGGGGGGGFGGGGGGGGGGGFEGEGFGGDCEF@:FGGGGGGGGGGG?FGCFGGGEFGGGGG
@M04277:241:000000000-B4VBL:1:1101:22673:2660;11066516;GCGACACA
ATTTTTAATACACCTGCTTCACGTCCCTATGTTGGGAAGTCCATATTTGTCTGCTTTTCTTGCAGCATCATTTCCT
+
CCCCCGGGFGGGD8C@C<EFFE@@C,@FFFCFFAFGGGGCGGGGGGGGDFGGGGFA<FGGFFGFGGGGGGGEGGGG
```

# Scripts for BD Rhapsody™ Data

https://bitbucket.org/CRSwDev/scripts-for-rhapsody

A software repository containing various tools to work with BD Rhapsody™ data and examples of how to interface with third-party packages.

- Data Conversion
- Downstream Analysis with Seurat and Scanpy
- Input Validation
- BD Rhapsody™ cell label details and python utility functions

# Pipeline Install Bundle (Docker-Free)

An experimental .tar.gz bundle to install the BD Rhapsody™ Sequence Analysis pipeline, without the use of a docker image. This contains cwl-runner and all the major required dependencies. Simply download, extract, and run.

**Download Link**

Linux only. Currently tested Linux versions:

- Ubuntu 16.04 / 20.04 / 22.04
- Red Hat 7
- CentOS 7 / 9

## Instructions

Extract the tar.gz bundle and enter that folder:

```
tar -xvzf rhapsodyPipeline-2.1.tar.gz
cd rhapsodyPipeline-2.1
```

To run the BD Rhapsody™ Sequence Analysis Pipeline:

First, define the input files and pipeline parameters in a pipeline_inputs.yml file. See the included "pipeline_inputs_template.yml" for instructions on formatting the YML file. Then, start the pipeline with this command:

```
./rhapsody pipeline --outdir results_dir pipeline_inputs.yml
```

To run a small test of the BD Rhapsody™ Sequence Analysis Pipeline with built-in demo data:

```
./rhapsody pipeline --outdir test_results test_files/test_smallDemo.yml
```

By default, the pipeline allows parallel node execution. To turn this off, use the --no-parallel option:

```
./rhapsody pipeline --no-parallel pipeline_inputs.yml
```

Any cwltool option can be added to the pipeline command. The YML file should always be the last argument. Examples are --outdir, --tmpdir-prefix, --leave-tmpdir.

```
./rhapsody pipeline --leave-tmpdir --outdir /path/to/directory pipeline_
inputs.yml
```

### Extra Utilities:

See additional instructions: Extra Utilities (page 120)

Given pairs of R1/R2 FASTQ files from BD Rhapsody™ libraries, only annotate the cell label and UMI of R1 and put it in the header of R2:

```
./rhapsody annotateCellLabelUmi inputs.yml
```

To create a new WTA Reference Archive for use as an input to the BD Rhapsody™ Sequence Analysis Pipeline:

```
./rhapsody makeRhapReference inputs.yml
```

To determine the amount of phiX contamination for a fastq file:

```
./rhapsody phiXContamination inputs.yml
```

Feedback welcome!

# 7

# References

## Software Packages in BD Rhapsody™ Sequence Analysis Pipeline

### anndata

Isaac Virshup et al. (2021). anndata: Annotated data. bioRxiv. doi: 10.1101/2021.12.16.473007.

### biopython

Cock et al. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics, Bioinformatics, Volume 25, Issue 11, June 2009, Pages 1422–1423, https://doi.org/10.1093/bioinformatics/btp163

### bowtie2

bowtie2: Fast and sensitive read alignment. [Software].
https://bowtie-bio.sourceforge.net/bowtie2/index.shtml

### bwa-mem2

Vasimuddin Md, Sanchit Misra, Heng Li, Srinivas Aluru. Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems. IEEE Parallel and Distributed Processing Symposium (IPDPS), 2019. 10.1109/IPDPS.2019.00041

### cython

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. Computing in Science & Engineering, 13(2), 31–39. https://ieeexplore.ieee.org/document/5582062

### htseq

Putri et al. (2022). Analysing high-throughput sequencing data in Python with HTSeq 2.0. Bioinformatics, btac166. https://doi.org/10.1093/bioinformatics/btac166

### IGBlast

Ye, J., Ma, N., Madden, T. L., & Ostell, J. M. (2013). IgBLAST: an immunoglobulin variable domain sequence analysis tool. Nucleic acids research, 41(Web Server issue), W34–W40. https://doi.org/10.1093/nar/gkt382

### jinja2

Pallets Projects. (2023). Jinja2. [Software]. https://palletsprojects.com/p/jinja/

## matplotlib

10. 4. Hunter. (2007). Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95. https://ieeexplore.ieee.org/document/4160265

### muon

Bredikhin, D., Kats, I. & Stegle, O. (2022). MUON: multimodal omics analysis framework. Genome Biology 2022 Feb 01. doi: 10.1186/s13059-021-02577-8.

### numba

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (pp. 1–6). https://numba.pydata.org/

### numpy

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. (2020).Array programming with NumPy. Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2

### opentsne

Pavlin G. Poličar, Martin Stražar and Blaž Zupan. (2019). openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. bioRxiv. https://www.biorxiv.org/content/10.1101/731877v3

### pandas

pandas development team. (2021). pandas-dev/pandas: Pandas (v2.1.0rc0). Zenodo. https://doi.org/10.5281/zenodo.605272

### pigz

The pigz developement team. (2023). pigz: A parallel implementation of gzip for modern multi-processor, multi-core machines. [Software]. https://zlib.net/pigz/

### pyir

Soto, C. et al. (2020). PyIR: a scalable wrapper for processing billions of immunoglobulin and T cell receptor sequences using IgBLAST. BMC Bioinformatics. https://github.com/crowelab/PyIR

### pysam

pysam: a Python module for reading and manipulating SAM/BAM files. [Software]. https://github.com/pysam-developers/pysam

### python-levenshtein

Max Bachmann. (2022). python-levenshtein. [Software]. https://github.com/maxbachmann/python-Levenshtein

## scikit-learn

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

## scipy

Virtanen et.al. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272. https://www.nature.com/articles/s41592-019-0686-2

## seaborn

Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021, https://doi.org/10.21105/joss.03021

## seqtk

Li, H. et al. (2023). Seqtk: a fast and lightweight tool for processing sequences in the FASTA or FASTQ format. [Software]. https://github.com/lh3/seqtk

## seurat

Hao, Y. et al. (2021). “Integrated analysis of multimodal single-cell data.” Cell. doi:10.1016/j.cell.2021.04.048, https://doi.org/10.1016/j.cell.2021.04.048.

## signac

Stuart, T. et al. (2021). "Single-cell chromatin state analysis with Signac." Nature Methods. doi:10.1038/s41592-021-01282-5, https://doi.org/10.1038/s41592-021-01282-5

## sinto

sinto: single-cell analysis tools. [Software]. https://github.com/timoast/sinto

## STAR

Dobin, A. et al. (2013). STAR: ultrafast universal RNA-seq aligner. Bioinformatics (Oxford, England), 29(1), 15–21. https://doi.org/10.1093/bioinformatics/bts635

## tensorflow-cpu

Martin et al. (2016). TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. 265–283.https://dl.acm.org/doi/10.5555/3026877.3026899

## Trinity

Grabherr, M. et al. (2011). Full-length transcriptome assembly from RNA-seq data without a reference genome. Nat Biotechnol. 2011 May 15;29(7):644-52.

# 8

# Release Notes

## v2.2 - March, 2024

### Added

- Added support for ATAC-Seq Assay and Multiomic ATAC-Seq Assay (WTA+ATAC-Seq)
- Added ability to customize STAR alignment parameters

### Updated

- Updated the immune cell type classifier to be more lenient in the percentage of bioproducts required to run
- Updated TCR BCR annotation software IGBlast to version 1.22
- Updated bead version detection

### Fixed

- Fixed error in dimensionality reduction when zero variable genes are found due to very sparse data

## v2.1 - Nov 10, 2023 (Internal and early access release only)

### Added

- Added TCR/BCR high-quality cell designation and associated metrics. This creates a new set of VDJ metrics similar to products where there is a putative cell call for VDJ libraries, separate from the cell call from associated gene expression libraries
- Added UMAP dimensionality reduction coordinates as an output file and also built those coordinates into the pipeline report, Seurat, and Scanpy outputs
- Added extra utility for only annotating the cell index and UMI of R1 and putting it in the header of R2
- Added support for Enhanced Cell Capture Beads V3

### Updated

- Updated Seurat output to separate mRNA and AbSeq data into the RNA and ADT assays respectively
- Updated Scanpy output to use Muon (.h5mu) and create mRNA and AbSeq data in separate anndata objects, rna and prot respectively

- Updated TCR/BCR dominant contigs file to include AIRR compliant germline columns
- Updated TCR/BCR dominant contigs file to only retain cell type appropriate chains. All chains are still available in the unfiltered contigs file.

- Updated TCR/BCR dominant contigs file to rename the column “duplicate_count” to “umi_count”, in accordance with AIRR’s definition update in v1.4.1
- Updated TCR/BCR dominant contig selection process, elevating the importance of a productive contig with high relative read count, and removing the CDR3 requirement
- Updated TCR/BCR DBEC algorithm to allow exceptions for CDR3 sequences not seen in any other cell, and CDR3 paired chains seen in other cells
- Updated TCR/BCR contig_id to correspond with annotated chain type

- Updated basic cell calling to scale better with small and large cell datasets, and prevent most inappropriately high cell calls derived from noise signatures
- Updated Alignment Category “No_Feature_Pct” metric to include targeted mRNA reads that are filtered out due to an invalid alignment
- Updated cell label annotation to improve the speed of annotation for reads with cell label sequences that contain more than 1 error
- Updated RAM requirements for VDJ_preprocess_reads on local server runs
- Updated error handling and reporting in read processing steps
- Updated logging to capture errors during alignment with STAR
- Updated FASTQ handling to skip reads with empty sequence
- Updated cell type classification model selection to better select an appropriate model when not all bioproducts are found in any one model
- Updated pipeline report to show sub-sampled tSNE and UMAP plots, in the case where the putative cell count exceeds 100,000
- Updated pipeline report to show details of refined cell calling, when refined cell calling is selected
- Updated bead version detection and read trimming

**Fixed**

- Fixed issue that caused failure when a gene symbol was named 'nan'
- Fixed issue with a quote mark in a gene symbol causing a failure in the Seurat output file generation
- Fixed rare division by zero issue in DBEC algorithm
- Fixed rare issue caused by including "SampleTag" in the Run_Name parameter

**Experimental**

- Added docker-free version of the pipeline, available for local server installs as a tar.gz bundle. Tested on Linux versions: Ubuntu 16 / 20 / 22 - Red Hat 7 - CentOS 7 / 9

**make_rhapsody_reference tool:**

- Added an “Extra_STAR_params” input to enable passing parameters to the STAR genomeGenerate process
- Updated to automatically generate a GTF for sequences added in the “Extra_sequences” FASTA input -- useful for transgenes

## v2.0 - June 14, 2023

- Major rewrite to read processing steps of the pipeline results in up to 7x faster performance and 2x less disk space required

- New cell-bioproduct datatable output file formats: MEX, for broad compatibility with downstream analysis tools. Seurat RDS and ScanPy H5AD for single files that include all cell metadata (i.e. Sample Tag, TCR/BCR, etc)
- Consolidated previously separate WTA and Targeted pipelines into one pipeline
- New updated WTA reference combines STAR index and matching GTF
- Built-in support for creating a new WTA reference with paired genome FASTA and GTF
- New Maximum_Threads parameter to limit the CPU usage on local server runs
- Basic cell caller is now the default algorithm. Refined cell calling algorithm can still be used by setting the Enable_Refined_Cell_Call parameter
- New pipeline input: Expected_Cell_Count - Guide the basic putative cell calling algorithm by providing an estimate of the number of cells expected. Usually this can be the number of cells loaded in the BD Rhapsody™ cartridge
- BAM files are not generated by default, but can be created using the Generate_Bam parameter
- Numerous other fixes and optimizations

## v1.12.1 - March 14, 2023

- Fix TCR pairing percent metrics

## v1.12 - Feb 21, 2023

- Added support for BD Rhapsody™ Enhanced bead V2 with an expanded cell label diversity
- Added support for Flex SMK (Sample Multiplexing Kit) allowing 24 species and cell type agnostic sample tags
- Upgraded CWL to version 1.2
- VDJ nodes are only executed when necessary
- Pipeline Report: Added cell label graph when an exact count is specified
- Added option to skip creating BAM file output
- Use productive status when collapsing chains for the VDJ perCell output file
- Dominant Contigs AIRR file now have DBEC filtering applied and are uncompressed. Both AIRR files have an additional column cell_type_experimental. The non-AIRR Dominant/Unfiltered files are no longer part of the pipeline output.
- Prioritize IG/TR gene features when annotating reads from a VDJ assay

## v1.11.1 - Dec 15, 2022

- Improved speed and disk usage of AnnotateReads step
- Update Pandas version to fix error: ValueError: Unstacked DataFrame is too big, causing int32 overflow
- Better prediction of RAM requirements
- Improved basic and refined putative cell calling algorithms
- Deletion of unnecessary intermediate files to save disk space
- Seven Bridges deployment: Fix for error Instance not available for automatic scheduling

## v1.11 - Aug 18, 2022

BD Rhapsody™ Targeted Analysis Pipeline and BD Rhapsody™ WTA Analysis Pipeline:

- Added a pipeline report HTML that contains information about the analysis including the metrics summary and graphs to visualize the results
- By default, reads aligned to exons and introns are now considered and represented in molecule counts. Added parameter to control this behavior.
- Added new "Alignment Categories" for TCR and BCR reads
- Added support for VDJ Adaptive Immune Receptor Repertoire (AIRR) standard format
- For pipeline run where putative cells are determined based on AbSeq (protein) counts, added file output of cell IDs corresponding to suspected protein aggregates
- Updated CWL workflow on Seven Bridges to fix memory failures and dynamically allocate resources for large datasets
- Improved flexibility for FASTQ file naming
- Updated Picard to version 2.27.4
- Updated bead version detection

## v1.10.1 - April 14, 2022

- Fixed issue with cell label detection on reads from TCR/BCR, when TCR/BCR libraries were combined with other library types (WTA, Targeted, AbSeq) in a single sequencing index.
- Fixed issue with processing FASTQ files whose filenames end in fq.gz

## v1.10 - January 24, 2022

BD Rhapsody™ Targeted Analysis Pipeline and BD Rhapsody™ WTA Analysis Pipeline:

- Updated VDJ pipeline with improved performance, new assembly algorithm, new metrics and new output files containing all available contig sequences
- Added support for BD Rhapsody™ Enhanced Beads, with automatic bead version detection
- Added option to call putative cells based on AbSeq read counts (for troubleshooting only)
- Added Alignment Categories section to metric summary which provides a breakdown of alignments for read pairs with a valid cell label and UMI
- Added separate metric summary files for each sample tag for experiments using BD Single-Cell Multiplexing kits
- Renamed various metrics in outputs to reflect multiomics nature of data
  (Target Type -> Bioproduct_Type, Gene/Target -> Bioproduct)
- Added Pct_Read_Pair_Overlap and Median Reads Per cell metric to metric summary
- Improved support for larger runs on SBG
- Updated workflow on SBG to improve editing of resource requirements
- Optimized pipeline metadata handling
- Improved checking of reference files

## v1.9.1 - October 6, 2020

BD Rhapsody™ WTA Analysis Pipeline:

- Improved putative cell calling algorithm to reduce overcalling of putative cells in high cell input experiments
- Updated alignment settings to improve AbSeq mapping when R2 read length is greater than 75 bases

## v1.9 - July 29, 2020

BD Rhapsody™ Targeted Analysis Pipeline and BD Rhapsody™ WTA Analysis Pipeline:

- Improved FASTQ file pairing - filenames are flexible and pairing is now based on read sequence identifier
- Optimized pipeline in various steps for memory and storage usage
- Fixed bugs related to Sample Multiplexing Kit noise and DBEC mean molecule metric

BD Rhapsody™ Targeted Analysis Pipeline:

- Support for BD Rhapsody™ VDJ CDR3 protocol
- Read and molecule counts for targets from same gene symbol are combined in the output tables
- Updated Bowtie2 alignment parameters for improved sensitivity

BD Rhapsody™ WTA Analysis Pipeline:

- Updated Pct_Cellular Metrics calculations to match Bioinformatics handbook descriptions
- Added support for supplemental reference fasta files, which allow alignment to transgenes, like viral RNA or GFP
- Updated STAR alignment parameters for improved sensitivity

## v1.8 - Oct 4, 2019

BD Rhapsody™ Targeted Analysis Pipeline and BD Rhapsody™ WTA Analysis Pipeline:

- Added Sample_Tag_ReadsPerCell.csv to Multiplex Output
- Optimized pipeline in various steps for memory usage
- Fixed bug in status determination for UMI_Adjusted_Stats.csv file

BD Rhapsody™ Targeted Analysis Pipeline:

- Updated Targets section in Metrics_Summary.csv to calculate metrics based on targets detected in putative cells only
- Removed Clustering Analysis and outputs

BD Rhapsody™ WTA Analysis Pipeline:

- Added support for BD® AbSeq libraries
- Removed Targets section in Metrics_Summary.csv for WTA only libraries
- Removed Pct_Error_Reads and Error_Depth in UMI_Adjusted_Stats.csv, which are not applicable to WTA only libraries

## v1.7.1 - August 7, 2019

- Added BD Rhapsody™ WTA Analysis Pipeline
- Fixed bug that can cause stalling when zero putative cells were identified
- Fixed bug that affected runs using Disable Refined Putative Cell Calling option

## v1.6.1 - July 2, 2019

- Increased memory limits for GetDataTable and Metrics
- Fixed bug associated with "No Multiplex" option on SBG
- Uses fewer resources in AddToSam step.

## v1.6 - June 10, 2019

- Added new options for putative cell determination:
  - Exact Cell Count: Set a specific number of cells as putative, based on those with the highest error-corrected read count
  - Disable Refined Putative Cell Calling: Determine putative cells using only the basic algorithm
- Updated to Python 3
- Updated alignment defaults (minor molecule count changes expected)
- Local install only - CWL files are bundled into one file

## v1.5 - March 14, 2019

- Added support for BD Single-cell multiplexing kit: Mouse Immune
- Updated various filtering thresholds to support sequencing runs with shorter read length
- Deprecated pipeline input: BAM input
- Fixed bug in Quality Filter (minor metrics changes expected)
- Optimized pipeline (computationally faster, more scalable to support larger input data size, and better logging)

## v1.3 - July 31, 2018

- Added support for BD® AbSeq assay
- Added support for BD® single-cell multiplexing kit - Mouse Immune
- New pipeline input - AbSeq Reference
- New pipeline outputs - Unfiltered cell-gene data tables
- Updated Metrics_Summary.csv to support metrics from multiple sequencing libraries
- Updated Recursive Substitution Error Correction (RSEC) algorithm (minor molecule count changes expected)
- Optimized pipeline to run faster

## v1.02 - Nov 27, 2017

- Added support for BD Single-cell multiplexing kit - Human
- Improved pipeline speed by deleting large temp files
- Removed network requirement when running locally
- bug fix for the wrong docker image name - Dec 13, 2017

**Becton, Dickinson and Company**
**BD Biosciences**
155 North McCarthy Boulevard
Milpitas, California 95035 USA

bdbiosciences.com

If you have any questions or suggestions, write an email to one of our support teams:

**Europe:** biox_support_emea@bd.com
**China:** bd-scm-bioinformatic@bd.com
**USA:** gmb-us-scm_bioinformatics@bd.com
**Rest of the world:** scomix@bd.com