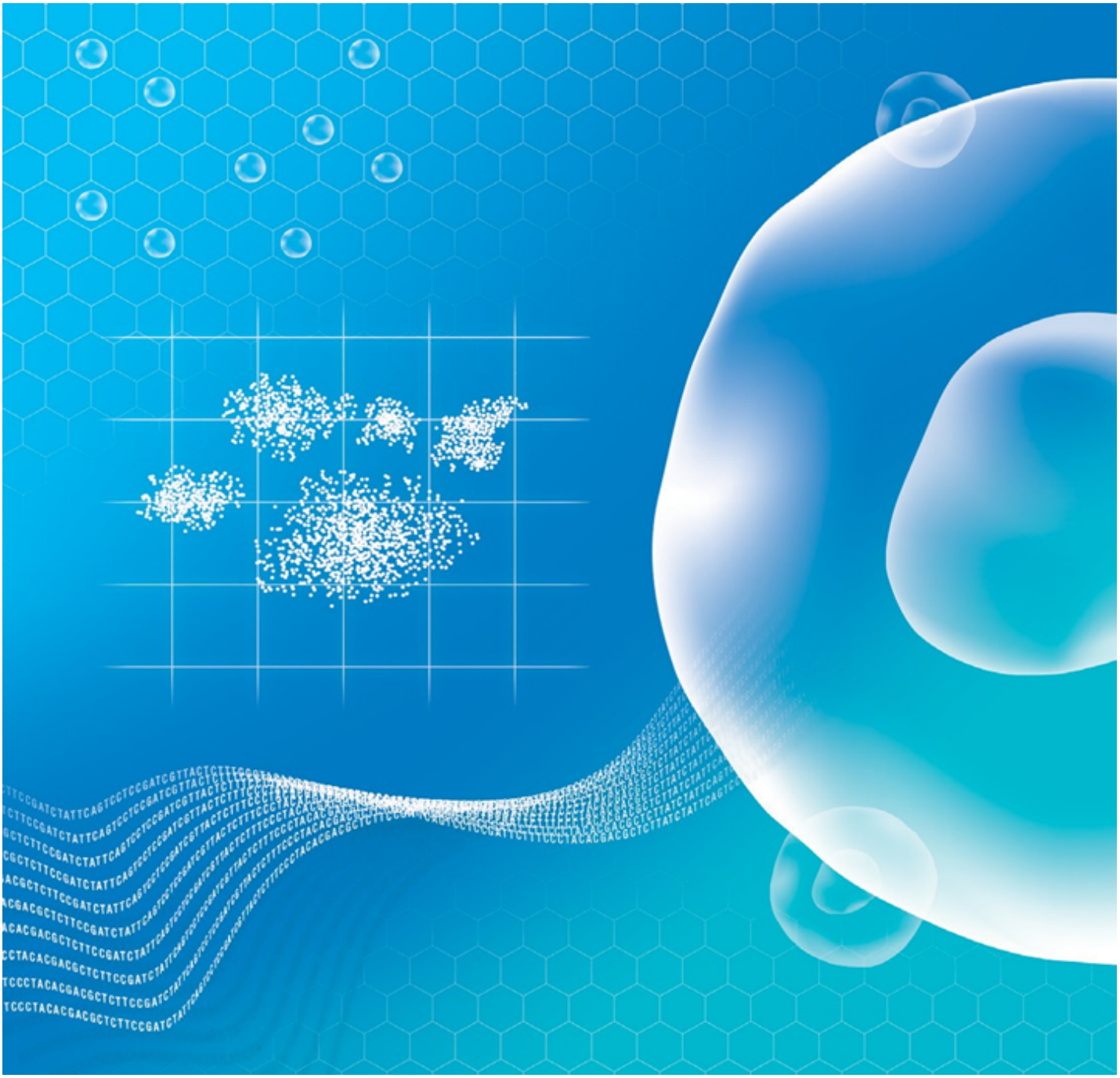


Omnition Analysis Software

User Guide

Version 1.0.1

BIO-RAD

Omnition Analysis Software

User Guide

Software Version 1.0.1

BIO-RAD

Bio-Rad Technical Support Department

The Bio-Rad Technical Support department in the U.S. is open Monday through Friday, 5:00 AM to 5:00 PM, Pacific time.

Phone: 1-800-424-6723, option 2

Email: Support@bio-rad.com (U.S./Canada Only)

For technical assistance outside the U.S. and Canada, contact your local technical support office or click the Contact Us link at www.bio-rad.com.

Legal Notice

Revision History

Document	Date	Description of Change
Omnition Analysis Software User Guide Bulletin #3408	December 2022	Revise the acceptable installation versions for Nextflow and update user guide to v1.0.1 (document update only)
Omnition Analysis Software User Guide	September 2022	New user guide, Omnition Analysis Software, ATAC Module

This page intentionally left blank.

Table of Contents

Chapter 1 Introduction

Omnition Analysis Software processes single-cell sequencing reads from libraries generated using the Bio-Rad SureCell ATAC-Seq Library Prep Kit,[1] and supports ATAC-seq protocols with combinatorial indexing, as described in Lareau *et al.*, Nature Biotechnology, 2019 (Lareau).

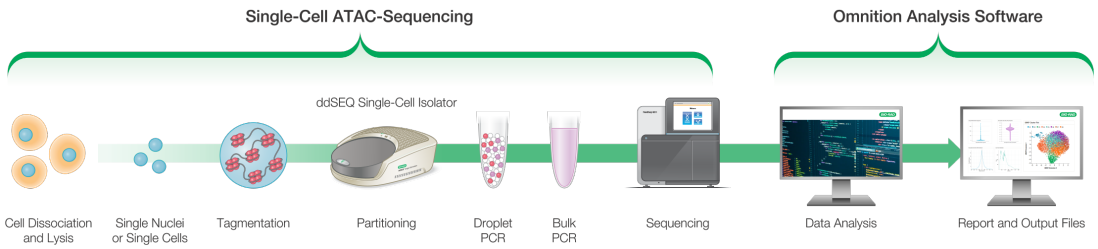
Omnition processes raw sequencing data into a format suitable for biological interpretation by

- Parsing and correcting single-cell barcodes
- Aligning reads to a reference genome
- Generating count matrices compatible with tertiary analysis tools

Omnition processes samples in batches, simplifying workflows and minimizing processing time for high throughput experiments. Pipeline runs are summarized in a single report that displays quality control metrics, alignment statistics, ATAC-seq metrics, and cell clustering for all samples included in the batch. If a combinatorial indexing protocol was used, the report displays samples grouped by their transposition indexes (TIs).

Fig. 1 illustrates the single-cell ATAC-seq analysis workflow used by Omnition Analysis Software.

Fig. 1: Bio-Rad SureCell ATAC-Seq workflow



The Omnition software is open-source and available at the following hyperlink:

http://github.com/BioRadOpenSource/omnition

[1] https://www.bio-rad.com/en-us/product/surecell-atac-seq-library-prep-kit?ID=PEXSR1MC1ORV

This page intentionally left blank.

Chapter 2 Pipeline Requirements and Verification

This section contains system and software requirements, as well as information to verify that Omnition Analysis Software works with its associated software components and can run a successful test workflow through the pipeline.

If the Omnition components are NOT already installed on the system, see Appendix C, Installation, for all installation procedures. You might require assistance from your system administrator to complete the installation.

If the components are already installed, no additional setup is required to launch the application. The Nextflow integrations with GitHub and Docker Hub orchestrate the workflow without any additional configuration, and you can immediately verify the pipeline functionality. For information, see Verifying the Pipeline on page 11.

System Requirements

Omnition is designed to run on a local Linux server, high performance computing (HPC) cluster, or cloud virtual machine, and has been tested on the 64-bit CentOS 7 and 8, and Ubuntu 18.04.6, 20.04 LTS, 21.04, and 21.10 Linux operating systems.

Important: Although they might be functional, Bio-Rad does not support additional Linux variants or other versions of the specified operating systems.

Bio-Rad recommends the minimum requirements specified in Table 1, although the system requirements and disk space scale according to the size of the input data.

Table 1. Recommended hardware requirements

Requirement	ATAC seq analyses	Combinatorial ATAC seq analyses
CPU	16	16
RAM	64 GB	128 GB

To identify resource requirements, benchmarks for RAM usage and processing time were determined from testing on an Amazon Web Services (AWS) cloud computing server. For information, see Performance Benchmarks on page 57.

Software Requirements

Omnition Analysis Software utilizes the Nextflow framework to connect individual processes, and runs the processes in virtual environments called containers using the Docker or Singularity container programs.

The applications are installed from the links specified below. For detailed installation instructions, see Setting Up the Environment Applications on page 61.

Important: The Nextflow framework and at least one of the container programs must be installed on the system before you can use Omnition.

Nextflow, v21.0.4 to v22.10.3

Note: You can use the Nextflow or Conda installation link below.

https://www.nextflow.io/docs/latest/getstarted.html#requirements

https://anaconda.org/bioconda/nextflow

Docker, v20.10.7 or later and/or Singularity v3.6.4 or later

Note: You can install both container applications, but only one is required.

https://docs.docker.com/get-docker/

https://docs.sylabs.io/guides/3.6/user-guide/quick_start.html#quick-installation-steps

Verifying the Pipeline

Omnition includes small demonstration datasets to verify that the environment has been properly built and all software dependencies are in place. These datasets do not generate biologically meaningful results and are meant only to verify correct environment configuration and software functionality for the end user.

To verify the success of the installation for each analysis type, run the Nextflow command for the container system that is installed on your computer (Singularity or Docker).

Important: Ensure there are no spaces after **demo_atac** and **demo_catac** in the **–profile** command line.

- To verify the atac-seq analysis pipeline, run the applicable command sequence shown below.

  Singularity:
  ```
  nextflow run BioRadOpenSource/omnition \
    -profile demo_atac,standard
  ```

  Docker:
  ```
  nextflow run BioRadOpenSource/omnition \
    -profile demo_atac,docker
  ```

- To verify the combinatorial atac-seq analysis pipeline, run the applicable command sequence shown below:

  Singularity:
  ```
  nextflow run BioRadOpenSource/omnition \
    -profile demo_catac,standard
  ```

  Docker:
  ```
  nextflow run BioRadOpenSource/omnition \
    -profile demo_catac,docker
  ```

This page intentionally left blank.

Chapter 3 Analysis Configurations

This chapter describes the configurations for ATAC-seq and combinatorial ATAC-seq analyses that are run in Omnition Analysis Software, and contains information on the following:

- File naming conventions
- Workflow structures
- Indexing configurations
- Workflow configuration overview and parameters
- Detailed demo configurations
- Pipeline outputs
- Appendices containing performance benchmarks, troubleshooting information, and installation procedures

Read Me First

The configuration and code examples used in this document are demonstration examples only. The directories and paths provided are meant to better illustrate the sequential workflows and configuration steps.

The directories and paths in your environment can be different and will correspond to the installed operating system and server/directory/folder names set up for your organization.

Important: This does not apply to the FASTQ naming convention for sample files. See FASTQ File Naming Conventions on page 14 for information.

FASTQ File Naming Conventions

Omnition Analysis Software accepts gzipped FASTQ files with the extensions .fastq.gz or .fq.gz. Omnition is designed to process files with names that follow the Illumina naming convention, as shown below:

SampleName_S1_L001_R1_001.fastq.gz

Important: Do not use hyphens in the SampleName, as they can interfere with how Omnition handles the splitting and merging of data within the workflow.

More information on the Illumina naming convention is available at the following hyperlink:

https://support.illumina.com/help/BaseSpace_OLH_009008/Content/Source/Informatics/BS/NamingConvention_FASTQ-files-swBS.htm

Omnition Workflow Types

Table 2 describes the workflow options available in Omnition Analysis Software. Omnition runs each workflow in accordance with the design and contents of the corresponding workflow configuration file. For information, see Workflow Configuration Overview and Parameters on page 24.

Table 2. Workflow Types

Type	Description
Reference	Omnition processes genome reference files to create the necessary dependencies for the analysis pipeline processes. The Reference Workflow is typically run once per reference version, and then the prepared reference is reused in subsequent analyses. See Reference Workflow on page 15.
Analysis	Omnition runs and analyzes FASTQ files in the pipeline, and produces reporting outputs. The Analysis Workflow requires a genome reference prepared by the Reference Workflow. See Analysis Workflow on page 17.
Full	The Full Workflow automatically runs the Analysis Workflow after the Reference Workflow run is finished. This workflow is included as a convenience for setup during the initial software deployment. See Full Workflow on page 17.

Reference Workflow

The Reference Workflow performs only the reference generation for the ATAC-seq and combinatorial ATAC-seq assays. Omnition officially supports human and mouse primary reference assemblies provided by ENSEMBL, and offers a mixed-species option to generate a “barnyard” reference.

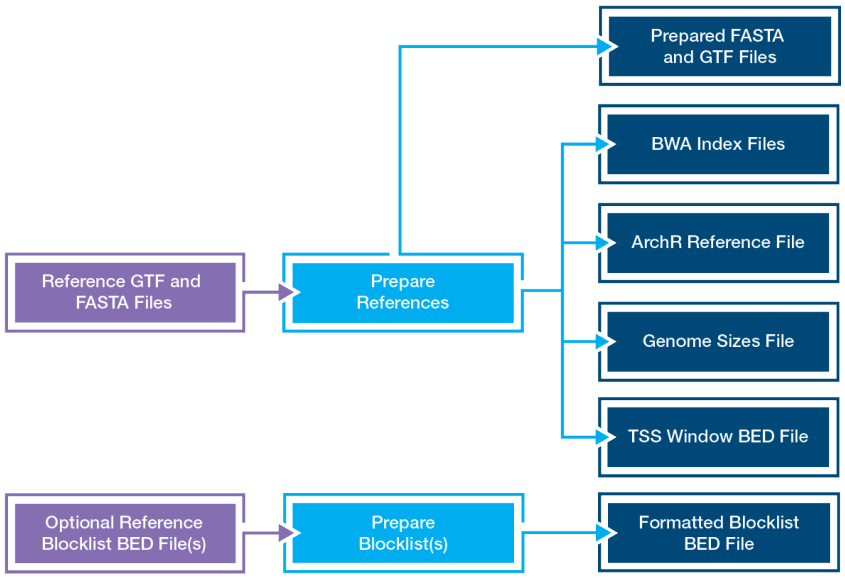
Note: The primary assembly represents the collection of assembled chromosomes, unlocalized and unplaced sequences that, when combined, should represent a non-redundant haploid genome. This excludes any of the alternate locus groups. You can obtain the primary assembly from ENSEMBL via the Download DNA sequence links as shown below:

Human: https://uswest.ensembl.org/Homo_sapiens/Info/Index

Mouse: https://useast.ensembl.org/Mus_musculus/Info/Index

For information on the parameters used to configure reference workflows, see Reference Workflow Parameters on page 25. To view demonstration Reference Workflows, see Reference Workflow: Single Species on page 34 and Reference Workflow: Mixed Species on page 37.

Fig. 2: ATAC-seq Reference Workflow



LEGEND

Purple	Inputs provided
Light blue	Processes performed
Dark blue	Output files produced

Input Files

The user is responsible for acquiring the appropriate input files for the Reference Workflow. Omnition requires the following:

- FASTA and GTF files must be obtained from ENSEMBL.
- Genome reference sequences must be formatted as FASTA files.
- Annotations must be formatted as GTF files.
- Sequence names in the FASTA and GTF files must match.

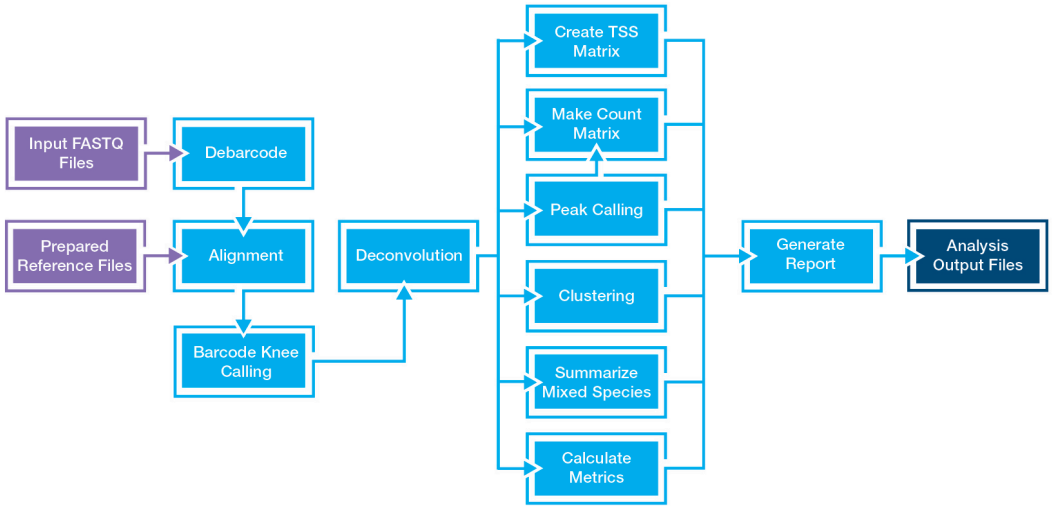
Note: Optionally, the input reference files can be compressed into gzip files.

Analysis Workflow

The Omnition Analysis Workflow processes paired-end FASTQ read files to generate outputs required for biological interpretation analysis. Each pipeline run also generates a quality control report that describes the analysis.

For information on parameters used to configure analysis workflows, see Analysis Workflow Parameters on page 26. To view demonstration Analysis Workflows, see Analysis Workflow Example: ATAC-Seq on page 40, Analysis Workflow Example: Combinatorial ATAC-Seq Superloading on page 44, and Analysis Workflow Example: Combinatorial ATAC-Seq Multiplexing on page 49.

Fig. 3: ATAC-seq Analysis Workflow Overview



LEGEND

Purple		Inputs provided
Light blue		Processes performed
Dark blue		Output files produced

Full Workflow

The Full Workflow is used to expedite processing during the initial deployment of Omnition. It runs the Reference Workflow first, followed automatically by the Analysis Workflow. For information on parameters used to configure full workflows, see Full Workflow Parameters on page 32.

Combinatorial Indexing Configurations

Combinatorial indexing permits superloading of cells within droplet partitions on the ddSEQ Single-Cell Isolator. Transposases used in combinatorial indexing ATAC-seq protocols carry indexes, and insert them into genomic DNA during the process of tagmentation.

When cells are superloaded, the probability of partitions with cell multiplets greatly increases; however, because the index carried by the transposase (TI, or transposition index) is present, it is possible to properly assign the fragments in partitions with cell multiplets to their cell of origin.

This section includes information to configure Omnition to process data accordingly for the following combinatorial indexing scenarios:

- Default TI Configuration (Superloading) on page 19
- Custom TI Configurations on page 20
- FASTQ-TI Configuration (Multiplexing) on page 22

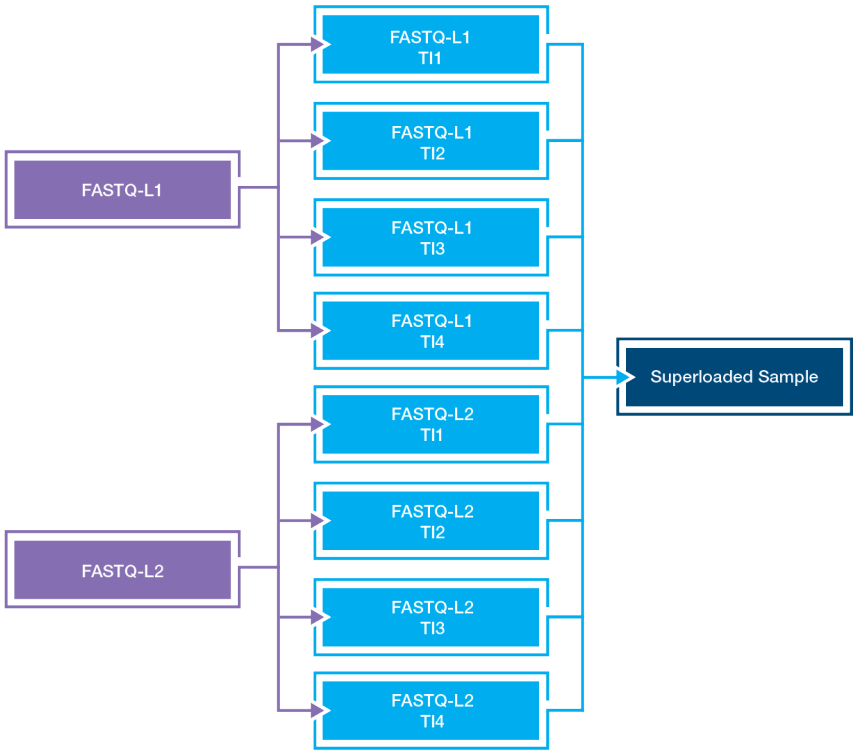
Default TI Configuration (Superloading)

By default, Omnition assumes the user is running a superloading experiment, such as processing a substantial number of FASTQ/TI combinations from a single origin sample. Combinatorial indexing allows deconvolution of cell multiplet partitions resulting from cell superloading on the ddSEQ Single-Cell isolator.

When analyzing a superloading experiment, Omnition merges TIs to create a single sample.

Fig. 4 illustrates a superloading experiment in which two FASTQ files (FASTQ-L1 and FASTQ-L2) contain four transposition indexes each (TI1 through TI4). Omnition combines the indexes to generate one sample, and produces one output report and one set of deliverable output files for the sample.

Fig. 4: Superloading experiment example



LEGEND

Purple		Inputs provided
Light blue		Processes performed
Dark blue		Output files produced

Custom TI Configurations

Important: Inaccuracies in combinatorial indexing configurations can lead to incorrect interpretation of data. Follow the guides in this section to ensure accurate analysis. For clarity, Omnition creates a file that reports the input configuration (sample_map.csv in the reports subdirectory).

The default settings in Omnition Analysis Software use the transposition indexes reported in Lareau. These experiments used 96 different hexamer TIs, which were sequenced as the first six bases following the bead barcode sequence in the first read of each pair.

In order to configure alternate TI sequences and locations, you can adjust the Analysis Workflow workflow configuration file using the **ti** parameter. For illustrations, see the workflow configurations (in YAML format) in Analysis Workflow Example: Combinatorial ATAC-Seq Superloading on page 44 and Analysis Workflow Example: Combinatorial ATAC-Seq Multiplexing on page 49.

Omnition supports TIs that are located as the first bases of Read 1 or Read 2. The following subsections contain accepted combinatorial indexing barcode formats.

Default Configuration: dsciATAC-seq (Lareau)

When running an analysis with the **barcodedTn5** parameter set to **true** and the **tiread** parameter set to **r1**, Omnition expects the following (as shown in the corresponding graphics):

- Read 1 contains the SureCell ATAC-seq barcode sequence, immediately followed by a 6-base transposition index and the mosaic end
- The remaining bases in Read 1 are from the DNA insert
- Read 2 contains only the DNA insert

Read 1: Bio-Rad SureCell ATAC-Seq with Combinatorial Index

Read 2: DNA insert only

Alternative Configuration: Read 2 Transposition Index

When running an analysis with the **barcodedTn5** parameter set to **true** and the **tiread** parameter set to **r2**, Omnition expects the following (as shown in the corresponding graphics):

- Read 1 contains the SureCell ATAC-seq barcode sequence.
- The remaining bases in Read 1 are from the DNA insert.
- The first bases in Read 2 are the transposition index.
- All remaining bases in Read 2 are from the DNA insert.

Read 1: Bio-Rad SureCell ATAC-Seq

Read 2: Transposition index + DNA insert

Alternative Configuration: i7 as TI

When running an analysis with the **barcodedTn5** parameter set to **true** and the **i7asti** parameter set to **true**, Omnition uses the i7 index sequence from each read name as the transposition index. No further configuration is required.

FASTQ-TI Configuration (Multiplexing)

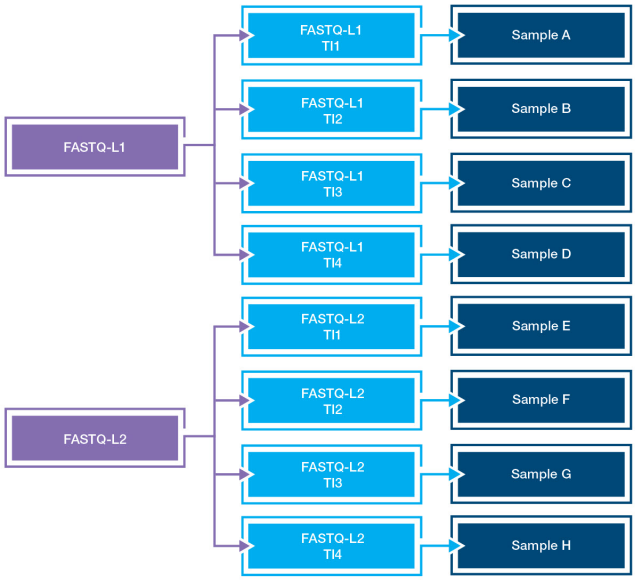
Combinatorial indexing allows cells from multiple samples to be deconvoluted from the same partition, and then assigned back to their sample of origin during data processing. It also offers flexibility in how samples are distributed across transposition indexes and FASTQ files.

Omnition supports combinatorial ATAC-seq multiplexing experiments, such as when the user pools distinct samples into one experiment, and each sample is labeled by one or more TIs during tagmentation. Multiplexing experiments also require an additional configuration file, in CSV format, to map each FASTQ-TI pair to a sample. For configuration information, see Analysis Workflow Example: Combinatorial ATAC-Seq Multiplexing on page 49.

Sample Multiplex Workflows

Fig. 5 illustrates a multiplex workflow and sample-index structure, in which two FASTQ files (FASTQ-L1 and FASTQ-L2) contain four transposition indexes each (TI1 through TI4). The indexes are mapped to eight distinct samples (Sample A through H). When the workflow is finished, Omnition produces one consolidated report containing eight samples, and eight sets of deliverable output files, one for each sample.

Fig. 5: Multiplex Workflow and Configuration CSV Example No. 1

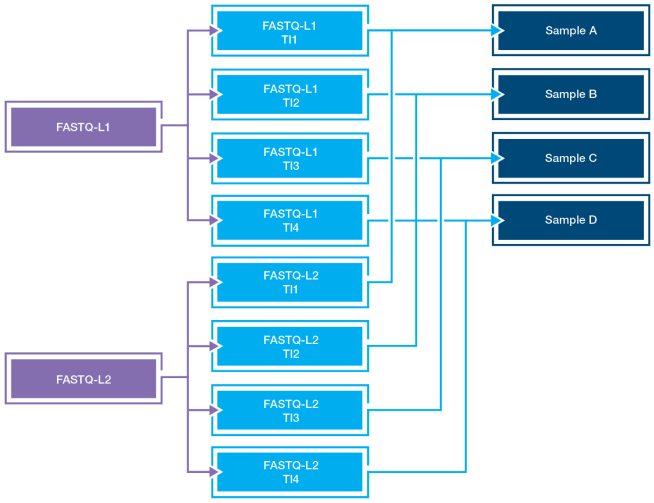


Sample	FASTQ	TI
Sample A	FASTQ-L1	TI1
Sample B	FASTQ-L1	TI2
Sample C	FASTQ-L1	TI3
Sample D	FASTQ-L1	TI4
Sample E	FASTQ-L2	TI1
Sample F	FASTQ-L2	TI2
Sample G	FASTQ-L2	TI3
Sample H	FASTQ-L2	TI4

Fig. 6 illustrates a second multiplex workflow and sample-index structure, in which two FASTQ files (FASTQ-L1 and FASTQ-L2) contain four transposition indexes each (TI1 through TI4). In this scenario, however, the combination of the FASTQ indexes (L1 and L2) and TIs (TI1 through TI4) are mapped to each of the four samples (A, B, C, and D) that have been split into two FASTQ files.

After the workflow is run, Omnition produces a single consolidated report containing four samples and four sets of deliverable output files, one for each sample.

Fig. 6: Multiplex Workflow and Configuration CSV Example No. 2



Sample	FASTQ	TI
Sample A	FASTQ-L1	TI1
Sample B	FASTQ-L1	TI2
Sample C	FASTQ-L1	TI3
Sample D	FASTQ-L1	TI4
Sample A	FASTQ-L2	TI1
Sample B	FASTQ-L2	TI2
Sample C	FASTQ-L2	TI3
Sample D	FASTQ-L2	TI4

LEGEND

Purple	Inputs provided
Light blue	Processes performed
Dark blue	Output files produced

Workflow Configuration Overview and Parameters

A configuration file must be set up for each workflow that you will run using Omnition.

Important: Examples included in this guide appear in YAML format. YAML is a data serialization language that is human readable and uses white-space indentation to indicate nesting in the formatted configuration file structure. Although JSON is also an accepted option, Bio-Rad recommends using YAML to facilitate construction and visualization of your configuration files based on the included examples. For more information, see yaml.org.

Each Omnition workflow configuration file begins with the unindented assay type parameter at the top level. Omnition supports ATAC-seq experiments, so you must use **atac** as the top-level parameter. In the following Reference Workflow example, applicable parameters are indented at different nesting levels below **atac**.

Important: Each nesting level is indented by four spaces.

```
atac:
    workflow: "reference"
    reference:
        directory: "/home/ubuntu/human-genome/"
        fasta: "/home/ubuntu/human-genome/human.fa.gz"
        gtf: "/home/ubuntu/human-genome/human.gtf.gz"
```

Table 3 explains how Omnition processes the parameter types used in the workflow configuration files.

Table 3. Parameter types

Type	Description
Global parameters	Rather than run a separate pipeline for each sample, Omnition processes the samples in a single batch during the run. Global parameters apply to all samples in the batch unless you include the **overrides** parameter in the analysis workflow configuration file. For information on all parameters you can use in a configuration file, see Reference Workflow Parameters on page 25, Analysis Workflow Parameters on page 26, and Full Workflow Parameters on page 32.
Sample overrides parameter	You can use the **overrides** parameter to change certain global analysis parameters to unique parameters, which are applicable only to certain samples in the batch (one or more). The **overrides** parameter is described in Analysis Workflow Parameters.

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Reference Workflow Parameters

Table 4 contains the configurable parameters for the Reference Workflow. All parameters in the Reference Workflow are global.

Table 4. Reference Workflow Parameters

Name	Type	Description	Default	Required?
directory *Nested under* reference	string	Path to a directory containing genome FASTA and GTF files	None	Yes
fasta *Nested under* reference	string	Path to the FASTA file	None	Yes
gtf *Nested under* reference	string	Path to the GTF file	None	Yes
Note: For both **fasta** and **gtf**, if you prepared a mixed species reference, you must provide a FASTA and GTF file for both species.				
tssWindowSize *Nested under* atac	integer	Integer for the size of the window around a transcription start site *Used for metric calculation.*	4000	No
mixed *Nested under* atac	boolean	Whether a mixed-species reference is being generated	false	No
blocklist *Nested under* reference	string	A path to a BED formatted blocklist file For information, see Blocklist on page 26.	None	No

Blocklist

Note: Bio-Rad has adopted the term "blocklist" to replace "blacklist."

A blocklist is used to mask regions of the reference genome during analysis.

Bio-Rad does not create or provide blocklists for use in Omnition Analysis Software. Blocklists that are compatible with Omnition are publicly available from the Boyle Laboratory at the following hyperlink:

https://github.com/Boyle-Lab/Blacklist

Note the following:

- If you provide a blocklist file, the chromosome names in the first column must match the chromosome names found in the annotation FASTA and GTF files.
- If you do not provide a blocklist file, Omnition generates an empty file, which serves as a required placeholder when the Analysis Workflow is run.

Analysis Workflow Parameters

By default, all parameters in the Analysis Workflow are global, applying to all samples in the batch, but per-sample configuration within a batch is available for a subset of parameters using the **overrides** parameter. For information, see overrides on page 31.

Important: The Analysis Workflow requires one or more genome references prepared by the Reference Workflow. You must provide the path to the reference directory as a parameter in the workflow configuration file for the applicable Analysis Workflow, as shown in the following example.

```
atac:
    workflow: "analysis"
    input: "/home/ubuntu/atac-reads/"
    reference:
        directory: "/home/ubuntu/mixed-genome/"
```

Table 5 on page 27 contains the configurable parameters for the Analysis Workflow.

Table 5. Analysis Workflow Parameters

Name	Type	Description	Default	Required?
input *Nested under* atac	string	Path to the directory containing FASTQ files for processing	None	Yes
reference *Nested under* atac	string	Path to the reference directory created by the Reference Workflow	None	Yes
fasta *Nested under* reference and directory	string	Path to the FASTA file	None	Yes
gtf *Nested under* reference and directory	string	Path to the GTF file	None	Yes
blocklist *Nested under* reference	string	Path to the user-provided blocklist OR the empty(placeholder) blocklist created by the Reference Workflow	None	Yes
mitoContig *Nested under* atac	string	Name of the mitochondrial contig in the genome reference	MT	No
mixed *Nested under* atac	boolean	Whether a mixed-species reference is generated	false	No
barcodedTn5 *Nested under* atac	boolean	Whether a combinatorial indexing workflow is run	false	No

Table 5. Analysis Workflow Parameters, continued

Name	Type	Description	Default	Required?
ti *Nested under* atac	list	TI = transposition index, used for tagmentation Indented list naming each TI sequence. For example: **ti1:** = **AAAGAA** **ti2:** = **AACAGC** **ti3:**...	Lareau	No
barcodedTn5config *Nested under* atac	string	Path to a **barcodedTn5** configuration file *This requires that* **barcodedTn5 = true**. *See* Custom TI Configurations on page 20	None	No
Important: If you are using a **barcodedTn5config** parameter in a multiplexing scenario, the names and formatting in the workflow configuration file and the FASTQ-TI configuration CSV file must match.				
i7asti *Nested under* atac	boolean	Whether to use the i7 index sequence as the combinatorial index *This requires that* **barcodedTn5 = true**.	False	No
tiread *Nested under* atac	string	The read that contains the TI sequence **r1** *or* **r2** *are accepted.* *This requires that* **barcodedTn5 = true** *and* **i7asti** = **false**.	r1	No
tssWindowSize *Nested under* atac	integer	Integer for the size of the window around the transcription start site	4000	No

Table 5. Analysis Workflow Parameters, continued

Name	Type	Description	Default	Required?
mergeMethod *Nested under* atac	string	The reads in a pair to use when determining the transposase insertion site that will be used for bead merging **r1**, **r2**, *or* **both** *are accepted.*	both	No
qualityThreshold *Nested under* atac	integer	Minimum mapping quality (MAPQ) for a read that will be used in analysis, downstream of alignment *Range can be* **0** *to* **255**.	30	No
barcode *Nested under* atac	N/A	Required to configure settings related to knee calling	N/A	No
force *Nested under* barcode	integer	When nested under **barcode**, controls the number of beads that pass bead filtration *When configured as a global parameter, all samples pass this number of beads through bead filtration.*	None	No
trim *Nested under* atac	integer	Number of bases to trim from the 5’ end of r2 reads	0	No

Table 5. Analysis Workflow Parameters, continued

Name	Type	Description	Default	Required?
sortSize *Nested under* atac	float	Controls the sort collection size in duplicate marking *Range can be* **0.1** to **1.0**. *Decreasing this value can remedy out-of-memory errors for samples with high duplication rates.*	0.25	No
rounding *Nested under* atac	integer	Rounds insert size to the nearest 10, 100, or 1000 bases before performing bead merging *If left at the default (0), no rounding is performed.*	0	No
maxInsertSize *Nested under* atac	integer	Largest insert size that the pipeline will recognize Must be more than **100**. *Fragments with inserts above this size are removed from the analysis after alignment.*	2000	No

Table 5. Analysis Workflow Parameters, continued

Name	Type	Description	Default	Required?
tierroroverride *Nested under* atac	boolean	Whether to ignore errors in the TI configuration file *This uses only the FASTQ-TI combinations listed in the barcoded Tn5config file.*	false	No
overrides *Nested under* atac	list	Indented list of sample parameters that are different from global batch parameters *You can use the **overrides** parameter to change the **barcode**, **trim**, and **mergeMethod** parameters, as follows:* ■ *per-sample (ATAC-seq)* ■ *per-FASTQ or FASTQ+TI (combinatorial ATAC-seq).* *See the Omnition README file in GitHub for information.*	None	No

Full Workflow Parameters

Table 6 contains the required parameters for the Full Workflow. You can also use all optional parameters described in the Reference Workflow and Analysis Workflow sections.

Table 6. Full Workflow Required Parameters

Name	Type	Description	Default	Required?
directory *Nested under* atac	string	Path to a directory containing genome FASTA and GTF files	None	Yes
fasta *Nested under* reference and directory	string	Path to the FASTA file	None	Yes
gtf *Nested under* reference and directory	string	Path to the GTF file	None	Yes
input *Nested under* atac	string	Path to the directory containing FASTQ files for processing	None	Yes

Example Configurations

This section provides common configurations for reference and analysis workflows, and uses demo directories and files in the command lines and configuration structures. You can practice using these scenarios, andd then adapt the structures to run workflows using your laboratory files and data.

Tip: Formatted versions of the configuration files shown in the examples are available from the Omnition repository in GitHub (https://github.com/BioRadOpenSource/omnition).

For information on demonstration Reference Workflows, see the following sections:

- Reference Workflow: Single Species on page 34
- Reference Workflow: Mixed Species on page 37

For information on demonstration Analysis Workflows, see the following sections:

- Analysis Workflow Example: ATAC-Seq on page 40
- Analysis Workflow Example: Combinatorial ATAC-Seq Superloading on page 44
- Analysis Workflow Example: Combinatorial ATAC-Seq Multiplexing on page 49

For information on output files, see Pipeline Outputs on page 55.

Reference Workflow: Single Species

In this example, the user has already conducted a single species experiment and wants to build a reference for human GRCh38.

To expedite processing, this demonstration uses a subset of the human and mouse genomes. Before analyzing actual experiments, you must generate a genome reference from the entire reference sequence (typically the primary assembly). For information, see Reference Workflow on page 15.

Prerequisites: Create and change to the human-genome directory

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

▶ Use the following commands to create and change to the **human-genome** directory.

```
mkdir /home/ubuntu/human-genome

cd /home/ubuntu/human-genome
```

Step 1: Download the GRCh38 genome reference materials

Important: The download commands in this step apply to the demo files only. When using your own files, retrieve them from the applicable storage location.

▶ From the **/home/ubuntu/human-genome** directory, use the following commands to download the genome reference materials from ENSEMBL.

```
wget "ftp://ftp.ensembl.org/pub/release-106/fasta/homo_sapiens/dna/Homo_
sapiens.GRCh38.dna.chromosome.10.fa.gz"

wget "ftp://ftp.ensembl.org/pub/release-106/fasta/homo_sapiens/dna/Homo_
sapiens.GRCh38.dna.chromosome.MT.fa.gz"

wget -O human.gtf.gz "ftp://ftp.ensembl.org/pub/release-106/gtf/homo_
sapiens/Homo_sapiens.GRCh38.106.gtf.gz"

cat Homo_sapiens.GRCh38.dna.chromosome.10.fa.gz \
Homo_sapiens.GRCh38.dna.chromosome.MT.fa.gz > human.fa.gz

rm *.chromosome.*.fa.gz
```

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Step 2: Create the workflow configuration file

The configuration file contains the assay name at the top level and indented parameters that describe the workflow type (reference) and file locations.

```
atac:
    workflow: "reference"
    reference:
        directory: "/home/ubuntu/human-genome/"
        fasta: "/home/ubuntu/human-genome/human.fa.gz"
        gtf: "/home/ubuntu/human-genome/human.gtf.gz"
```

To create and save the file

1. Complete the following substeps to create the configuration file, as shown in the above example.
    a. After the assay name, enter the **reference** parameter.
    b. Below the **reference** parameter, enter the **directory** parameter, with the path to the reference directory, and then enter the **fasta** and **gtf** parameters, with paths to the file sets downloaded from ENSEMBL.
2. Save the file as **reference-human.yaml** in the reference directory.

Step 3: Run the Reference Workflow

▶ Use the following Nextflow command to run the Reference Workflow.

```
nextflow run BioRadOpenSource/omnition \
  -params-file reference-human.yaml \
  -profile standard
```

Step 4: Verify the outputs

As the Reference Workflow runs, a progress report appears on the screen.

```
--------------------------------------------------------------------------------

        ###################################################################
      ###################################################################
    ###################################################################
   #########          #    ##         ######          ####     ###         #######
   #########   ###    ##   #    ####    #####   ####   ###      ###   ###    ######
   ########          ##   #   #####    #            ###   ##   #   #####   ######
   ########   ####   #    #   ####    #####   ####   #          #   ####   #######
   #######          ##   ###        #######   ####      #####             ########
    ###################################################################
      ###################################################################
        ###################################################################
                            Single-cell Software

--------------------------------------------------------------------------------


INFO: fasta and gtf match
INFO: [ATAC] Executing reference workflow.
executor >  local (10)
[95/0ba114] process > PULL_IMAGES:SINGULARITY_PULL              [100%] 1 of 1 ✔
[5b/03e40d] process > ATAC_REFERENCE:GUNZIP_FASTA (human.fa.gz) [100%] 1 of 1 ✔
[1c/b51a8a] process > ATAC_REFERENCE:GUNZIP_GTF (human.gtf.gz)  [100%] 1 of 1 ✔
[43/e2ec62] process > ATAC_REFERENCE:GENERATE_EMPTY_BLOCKLIST   [100%] 1 of 1 ✔
[e8/69dd4e] process > ATAC_REFERENCE:FILTER_REFERENCES          [100%] 1 of 1 ✔
[48/208d5a] process > ATAC_REFERENCE:FILTER_BLOCKLISTS          [100%] 1 of 1 ✔
[c3/404e27] process > ATAC_REFERENCE:BWA_INDEX                  [100%] 1 of 1 ✔
[9d/503bf6] process > ATAC_REFERENCE:ARCHR_REFERENCE            [100%] 1 of 1 ✔
[9b/2cca5e] process > ATAC_REFERENCE:GENERATE_GENOME_SIZES      [100%] 1 of 1 ✔
[ea/e36a72] process > ATAC_REFERENCE:GENERATE_TSS_WINDOWS       [100%] 1 of 1 ✔
Completed at: 08-Jun-2022 18:54:35
Duration    : 2m 10s
CPU hours   : 0.1
Succeeded   : 10
```

When the Reference Workflow run is finished, the reference output directory should contain the files shown below.

▶ To view the files, run the `ls` command.

```
(base) ubuntu@ip-10-0-0-231 ~/human-genome $ ls -1
TSS.4000.bed
archr
bwa-index
filtered.blocklist.bed
genome.sizes
human.fa.gz
human.filtered.gtf
human.gtf.gz
reference.yaml
results
work
```

Reference Workflow: Mixed Species

In this example, the user has already conducted a human-mouse mixed-species experiment and wants to build a reference for human GRCh38 and mouse GRCm39.

To expedite processing, this demonstration uses a subset of the human and mouse genomes. Before analyzing actual experiments, you must generate a genome reference from the entire reference sequence (typically the primary assembly). For information, see Reference Workflow on page 15.

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Prerequisites: Create and change to the mixed-genome directory

▶ Use the following commands to create and change to the **mixed-genome** directory.

```
mkdir /home/ubuntu/mixed-genome

cd /home/ubuntu/mixed-genome
```

Step 1: Download the GRCh38 and GRCh39 genome reference materials

Important: The download commands in this step apply to the demo files only. When using your own files, retrieve them from the applicable storage location.

▶ From the **/home/ubuntu/mixed-genome** directory, use the following commands to download the genome reference materials from ENSEMBL.

```
wget "ftp://ftp.ensembl.org/pub/release-106/fasta/homo_sapiens/dna/Homo_
sapiens.GRCh38.dna.chromosome.10.fa.gz"

wget "ftp://ftp.ensembl.org/pub/release-106/fasta/homo_sapiens/dna/Homo_
sapiens.GRCh38.dna.chromosome.MT.fa.gz"

wget -O human.gtf.gz "ftp://ftp.ensembl.org/pub/release-106/gtf/homo_
sapiens/Homo_sapiens.GRCh38.106.gtf.gz"

cat Homo_sapiens.GRCh38.dna.chromosome.10.fa.gz \
Homo_sapiens.GRCh38.dna.chromosome.MT.fa.gz > human.fa.gz

wget "ftp://ftp.ensembl.org/pub/release-106/fasta/mus_musculus/dna/Mus_
musculus.GRCm39.dna.chromosome.10.fa.gz"

wget "ftp://ftp.ensembl.org/pub/release-106/fasta/mus_musculus/dna/Mus_
musculus.GRCm39.dna.chromosome.MT.fa.gz"

cat Mus_musculus.GRCm39.dna.chromosome.10.fa.gz \
Mus_musculus.GRCm39.dna.chromosome.MT.fa.gz > mouse.fa.gz

rm *.chromosome.*.fa.gz
```

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Step 2: Create workflow configuration file

The configuration file contains the assay name at the top level and indented parameters that describe the workflow type (reference) and file locations.

```
atac:
    workflow: "reference"
    reference:
        directory: "/home/ubuntu/mixed-genome/"
        fasta:
            species1: "/home/ubuntu/mixed-genome/human.fa.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.fa.gz"
        gtf:
            species1: "/home/ubuntu/mixed-genome/human.gtf.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.gtf.gz"
    mixed: true
```

To create and save the file

1. Complete the following substeps to create the configuration file, as shown in the above example.
    a. After the assay name, enter the **reference** parameter.
    b. Below the **reference** parameter, enter the **directory** parameter, with the path to the reference directory, and then enter the **fasta** and **gtf** parameters, with paths to the file sets downloaded from ENSEMBL.

       Note: For mixed species workflows, you must specify a file for each species under **fasta** and **gtf**.

    c. Set the **mixed** parameter to **true**.
2. Save the file as **reference-mixed.yaml** in the reference directory.

Step 3: Run the Reference Workflow

▶ Use the following Nextflow command to run the Reference Workflow.

```
nextflow run BioRadOpenSource/omnition \
  -params-file reference-mixed.yaml \
  -profile standard
```

Step 4: Verify the outputs

As the Reference Workflow runs, a progress report appears on the screen.

```
--------------------------------------------------------------------------------

        ###################################################################
      ###################################################################
    ###################################################################
   #########          #    ##         ######          ####     ###         #######
   #########   ###    ##   #    ####    #####   ####   ###      ###   ###    ######
   ########          ##   #   #####    #            ###   ##   #   #####   ######
   ########   ####   #    #   ####    #####   ####   #          #   ####   #######
   #######          ##   ###        #######   ####      #####             ########
    ###################################################################
      ###################################################################
        ###################################################################
                            Single-cell Software

--------------------------------------------------------------------------------


INFO: [ATAC] Executing mixed species workflow.
INFO: fasta and gtf match
INFO: [ATAC] Executing reference workflow.
executor >  local (13)
[fa/a92b38] process > PULL_IMAGES:SINGULARITY_PULL              [100%] 1 of 1 ✔
[ac/0ece67] process > ATAC_REFERENCE:GUNZIP_FASTA (human.fa.gz) [100%] 2 of 2 ✔
[2b/022e89] process > ATAC_REFERENCE:GUNZIP_GTF (human.gtf.gz)  [100%] 2 of 2 ✔
[db/a55881] process > ATAC_REFERENCE:COMBINE_REFERENCES         [100%] 1 of 1 ✔
[75/8e6d29] process > ATAC_REFERENCE:GENERATE_EMPTY_BLOCKLIST   [100%] 1 of 1 ✔
[65/5ef97f] process > ATAC_REFERENCE:FILTER_REFERENCES          [100%] 1 of 1 ✔
[78/5c2186] process > ATAC_REFERENCE:FILTER_BLOCKLISTS          [100%] 1 of 1 ✔
[d6/e96a56] process > ATAC_REFERENCE:BWA_INDEX                  [100%] 1 of 1 ✔
[23/f7702e] process > ATAC_REFERENCE:ARCHR_REFERENCE            [100%] 1 of 1 ✔
[0c/c319ff] process > ATAC_REFERENCE:GENERATE_GENOME_SIZES       [100%] 1 of 1 ✔
[37/38c862] process > ATAC_REFERENCE:GENERATE_TSS_WINDOWS       [100%] 1 of 1 ✔
Completed at: 08-Jun-2022 18:30:48
Duration    : 4m 44s
CPU hours   : 0.2
Succeeded   : 13
```

When the Reference Workflow run is finished, the reference output directory should contain the files shown below.

▶ To view the files, run the `ls` command.

```
(base) ubuntu@ip-10-0-0-231 ~/mixed-genome $ ls -1
TSS.4000.bed
archr
bwa-index
filtered.blocklist.bed
genome.sizes
human.fa.gz
human.gtf.gz
mixed.fa
mixed.filtered.gtf
mixed.gtf
mouse.fa.gz
mouse.gtf.gz
reference.yaml
results
work
```

Analysis Workflow Example: ATAC-Seq

This example uses a mixed-species experiment in which the user has already done the following:

- Generated sequencing data using the SureCell ATAC-Seq Library Prep Kit from Bio-Rad.
- Prepared a genome reference from Reference Workflow: Mixed Species on page 37 and saved it to the **/home/ubuntu/mixed-reference** directory.

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Prerequisite: Create and change to the atac-reads directory

▶ Use the following commands to create and change to the **atac-reads** directory.

Note: The reads in this demonstration are paired-end reads that were not lane split during demultiplexing. If lane split reads are provided, Omnition merges them during analysis.

```
mkdir /home/ubuntu/atac-reads

cd /home/ubuntu/atac-reads
```

Step 1: Download the FASTQ files to the atac-reads directory

Important: The download commands in this step apply to the demo files only. When using your own files, retrieve them from the applicable storage location.

▶ From the **/home/ubuntu/atac-reads** directory, use the following commands to download the demo FASTQ files.

```
cp ~/.nextflow/assets/BioRadOpenSource/omnition/test/data/atac/normal/
*.fastq.gz ./
```

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Step 2: Create the workflow configuration file

The configuration file contains the assay name at the top level and indented parameters describe the workflow type (analysis).

```
atac:
    workflow: "analysis"
    input: "/home/ubuntu/atac-reads/"
    reference:
        directory: "/home/ubuntu/mixed-genome/"
        fasta:
            species1: "/home/ubuntu/mixed-genome/human.fa.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.fa.gz"
        gtf:
            species1: "/home/ubuntu/mixed-genome/human.gtf.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.gtf.gz"
        blocklist: "/home/ubuntu/mixed-genome/filtered.blocklist.bed"
    mixed: true
```

To create and save the file

1. Complete the following substeps to create the configuration file, as shown in the above example.
    a. Enter the assay type parameter (**atac**) and the workflow parameter (**analysis**).
    b. Below the **workflow** parameter, enter the **input** parameter, which is the path to the FASTQ files for the analysis run.
    c. Below the **input** parameter, enter the **reference** parameter.
    d. Below the **reference** parameter, enter the **directory** parameter, with the path to the reference directory, and then enter the **fasta** and **gtf** parameters, with paths to the file sets downloaded from ENSEMBL.

       Note: For mixed species workflows, you must specify a file for each species under **fasta** and **gtf**.
    e. To add a **blocklist** parameter, enter **blocklist** and then enter the path to the blocklist file.
    f. To indicate a mixed species experiment, set the **mixed** parameter to **true**.
2. Save the file as **analysis-mixed.yaml** in the **/home/ubuntu/atac-reads** directory.

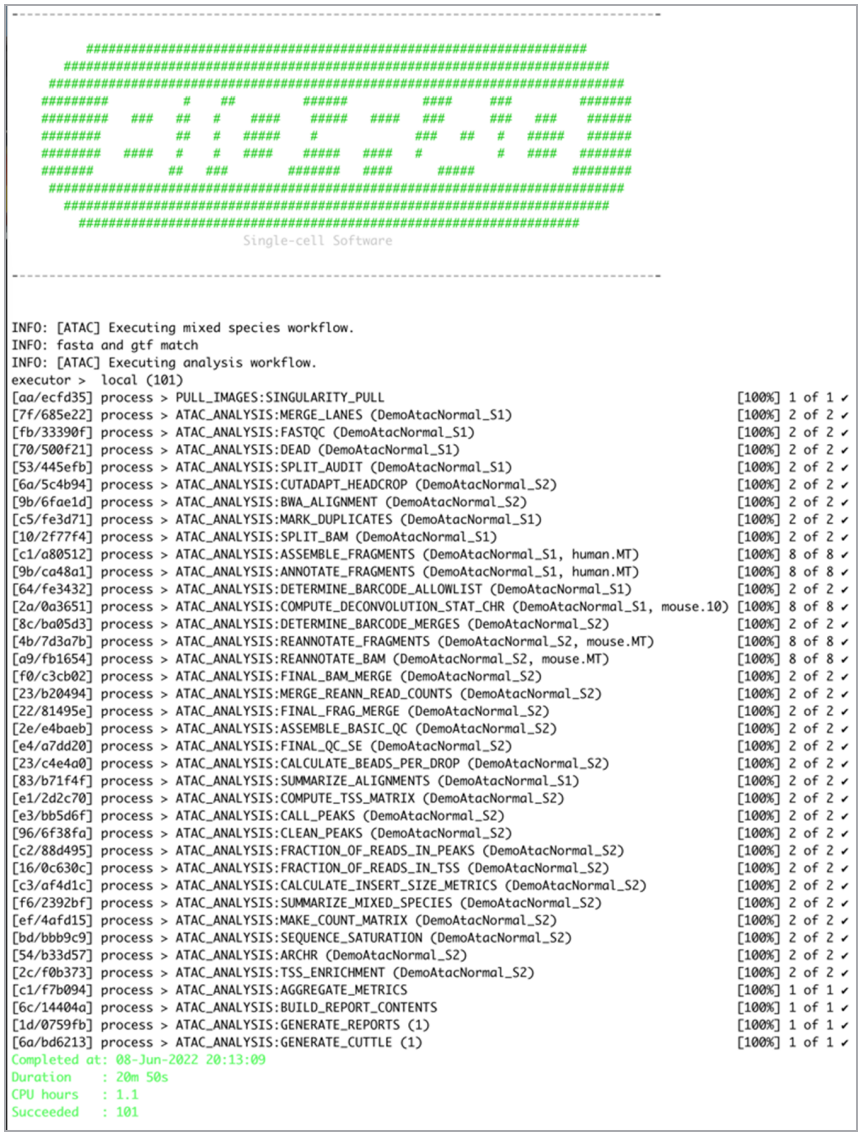
Step 3: Run the Analysis Workflow

▶ Use the following Nextflow command to run the Analysis Workflow:

```
nextflow run BioRadOpenSource/omnition \
  -params-file analysis-mixed.yaml \
  -profile standard
```

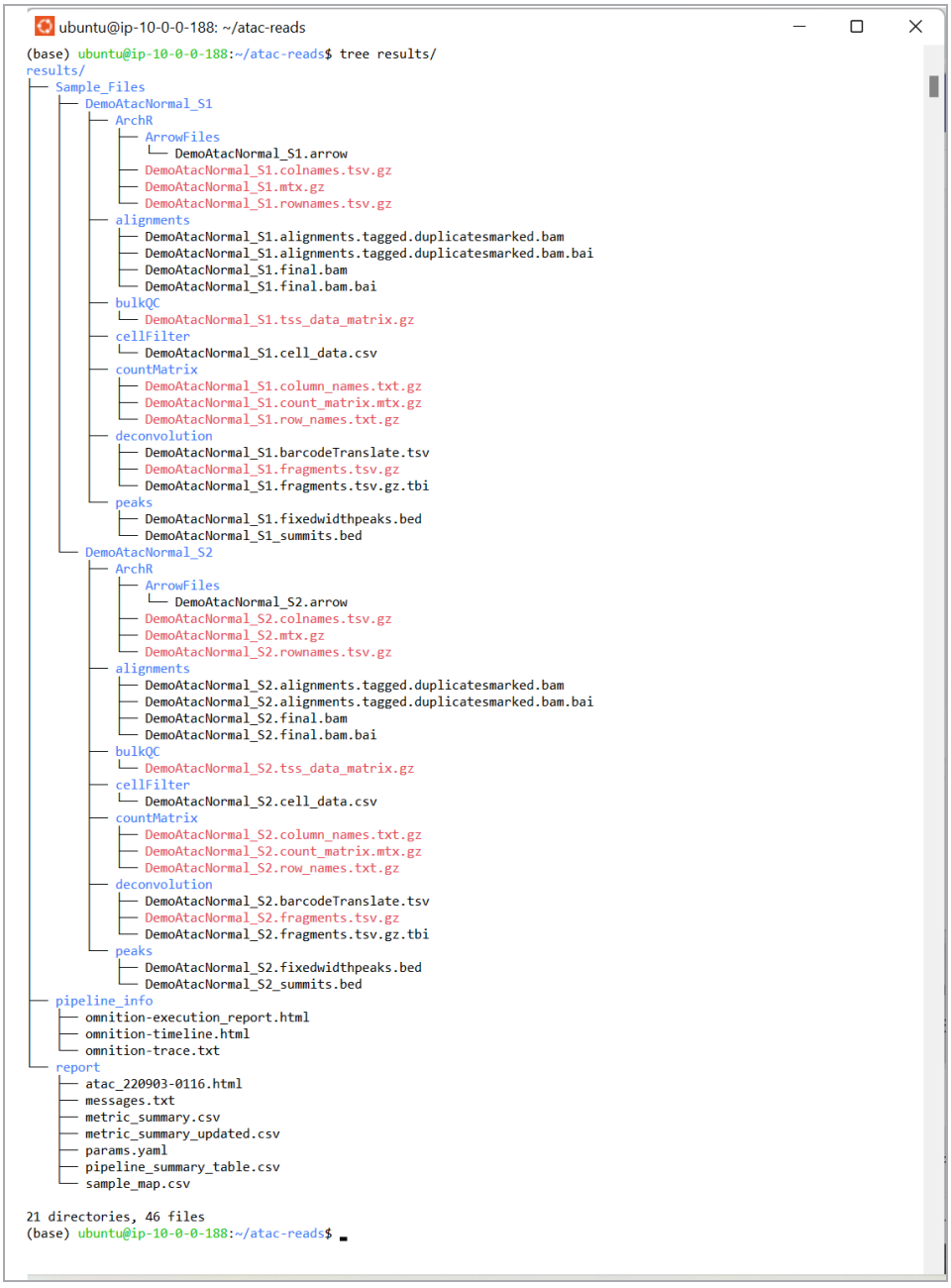
Step 4: Verify the outputs

As the Analysis Workflow runs, a progress report appears on the screen.



When the Analysis Workflow run is finished, the output directory should contain the files shown below.

▶ To view the file list, run the `tree` command.



Analysis Workflow Example: Combinatorial ATAC-Seq Superloading

This example uses a mixed-species experiment in which the user has already done the following:

- Generated sequencing data using the SureCell ATAC-Seq Library Prep Kit from Bio-Rad.
- Used a set of indexed transposases for nuclear transposition.
- Prepared a genome reference from Reference Workflow: Mixed Species on page 37 and saved it to the **/home/ubuntu/mixed-reference** directory.

A combinatorial ATAC-seq superloading analysis requires additional configuration. The preset default values in the Omnition configuration assume that you have conducted an experiment using all 96 transposition indexes, as in Lareau. The transposition index sequence makes up the first six bases following the bead barcode sequence in the first read of each read pair.

Note: This example uses a subset of the Lareau transposition indexes, which are defined in the configuration file.

The reads in this demonstration are paired-end reads that were not lane split during demultiplexing. If lane split reads are provided, Omnition merges them during analysis.

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Prerequisite: Create and change to the combinatorial-atac-reads directory

▶ Use the following commands to create and change to the **combinatorial-atac-reads** directory.

```
mkdir /home/ubuntu/combinatorial-atac-reads

cd /home/ubuntu/combinatorial-atac-reads
```

Step 1: Download the FASTQ files to the combinatorial-atac-reads directory

Important: The download commands in this step apply to the demo files only. When using your own files, retrieve them from the applicable storage location.

▶ From the **/home/ubuntu/combinatorial-atac-reads** directory, use the following commands to download the demonstration FASTQ files

```
cp ~/.nextflow/assets/BioRadOpenSource/omnition/test/data/atac/combinatorial/
*.fastq.gz ./
```

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Step 2: Create the workflow and indexing configuration file

The configuration file contains the assay name at the top level and indented parameters describe the workflow type (analysis).

```
atac:
    workflow: "analysis"
    input: "/home/ubuntu/combinatorial-atac-reads"
    reference:
        directory: "/home/ubuntu/mixed-genome"
        fasta:
            species1: "/home/ubuntu/mixed-genome/human.fa.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.fa.gz"
        gtf:
            species1: "/home/ubuntu/mixed-genome/human.gtf.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.gtf.gz"
        blocklist: "/home/ubuntu/mixed-genome/filtered.blocklist.bed"
    mixed: true
    barcodedTn5: true
    ti:
        ti1: "AAAGAA"
        ti2: "TTTGGG"
    tiread: "r1"
```

To create and save the file

1. Complete the following substeps to create the configuration file, as shown in the above example.
    a. Enter the assay type parameter (**atac**) and the workflow parameter (**analysis**).
    b. Below the **workflow** parameter, enter the **input** parameter, which is the path to the FASTQ files for the analysis run.
    c. Below the **input** parameter, enter the **reference** parameter.
    d. Below the **reference** parameter, enter the **directory** parameter, with the path to the reference directory, and then enter the **fasta** and **gtf** parameters, with paths to the file sets downloaded from ENSEMBL.

       Note: For mixed species workflows, you must specify a file for each species under **fasta** and **gtf**.

e. To add a **blocklist** parameter, enter **blocklist** and then enter the path to the blocklist file.

f. To indicate a mixed species experiment, set the **mixed** parameter to **true**.

g. To indicate a combinatorial indexing workflow, set the **barcodedTn5** parameter to **true**.

h. Below **barcodedTn5**, enter the **ti** parameter, and then enter the specific TIs.

Note: This example defines the TIs as part of a list and tells Omnition that the TI is the first six bases following the barcode in the first read pair.

i. Enter the **tiread** parameter and specify the read.

2. Save the file as **analysis-superloading.yaml** in the **/home/ubuntu/combinatorial-atac-reads** directory.

Step 3: Run the Analysis Workflow

▶ Run the following command in Nextflow to run the Analysis Workflow.

```
nextflow run BioRadOpenSource/omnition \
  -params-file analysis-superloading.yaml \
  -profile standard
```

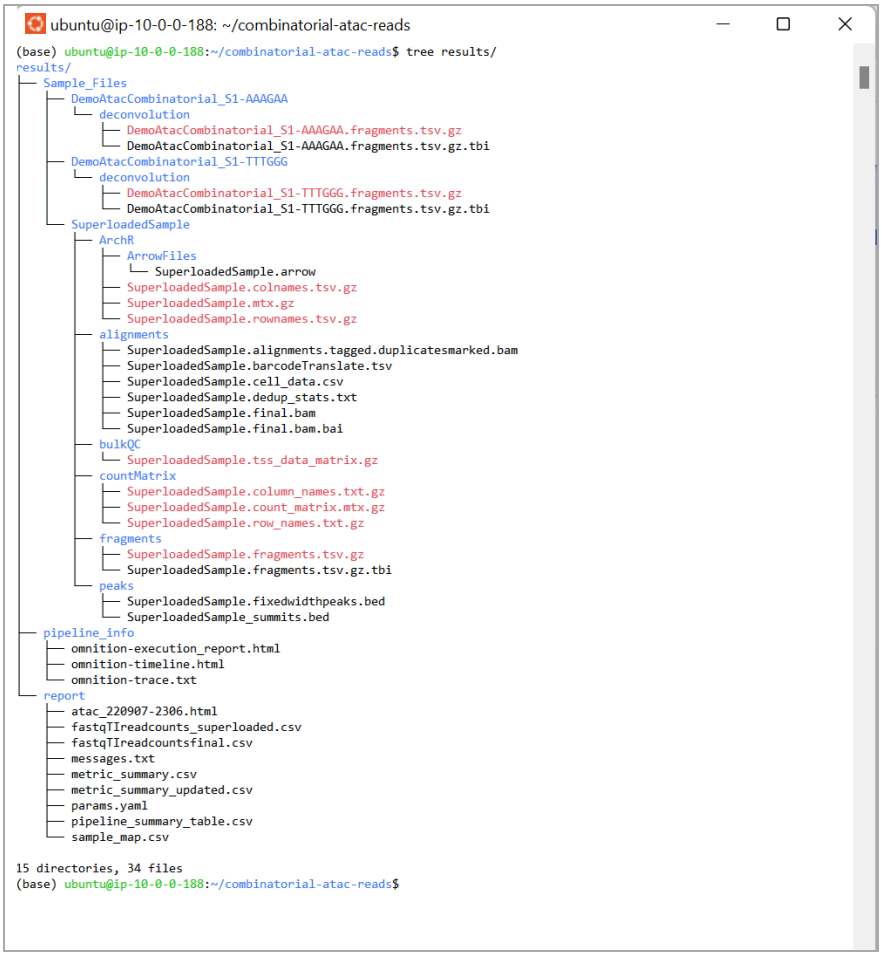
Step 4: Verify the outputs

As the Analysis Workflow runs, a progress report appears on the screen.

```
--------------------------------------------------------------------------------

        ############################################################
      ################################################################
    ####################################################################
   #########         #    ##        ######         ####    ###        #######
   #########   ###   ##   #    ####    #####   ####   ###     ###   ###    ######
   ########          ##   #   #####    #           ###   ##   #   #####   ######
   ########   ####   #    #   ####    #####   ####   #         #   ####   #######
   #######          ##   ###        #######   ####      #####            ########
    ####################################################################
      ################################################################
        ############################################################
                              Single-cell Software

--------------------------------------------------------------------------------


INFO: [ATAC] Executing mixed species workflow.
INFO: fasta and gtf match
INFO: [ATAC] Executing analysis workflow.
INFO: [ATAC] Executing barcodedTn5 workflow.
INFO: [ATAC] TIs are set to be located in r1.
INFO: [ATAC] No 'BarcodedTn5Config' Config file provided,                    executing superloading workflow.
executor >  local (92)
[5e/0cee6f] process > PULL_IMAGES:SINGULARITY_PULL                                                                [100%] 1 of 1 ✔
[96/f0f6fa] process > ATAC_ANALYSIS:MERGE_LANES (DemoAtacCombinatorial_S1)                                        [100%] 1 of 1 ✔
[72/73d124] process > ATAC_ANALYSIS:FASTQC (DemoAtacCombinatorial_S1)                                             [100%] 1 of 1 ✔
[f8/e7b899] process > ATAC_ANALYSIS:TI_DEAD_CONFIG                                                               [100%] 1 of 1 ✔
[ef/db29db] process > ATAC_ANALYSIS:DEAD (DemoAtacCombinatorial_S1)                                              [100%] 1 of 1 ✔
[1c/483eb2] process > ATAC_ANALYSIS:SPLIT_AUDIT (DemoAtacCombinatorial_S1)                                       [100%] 1 of 1 ✔
[1b/6dd71f] process > ATAC_ANALYSIS:CHECK_TI_COUNTS_SUPERLOADED (DemoAtacCombinatorial_S1)                       [100%] 1 of 1 ✔
[93/a10cea] process > ATAC_ANALYSIS:TI_WARNING_MESSAGES (1)                                                      [100%] 1 of 1 ✔
[9c/7e5361] process > ATAC_ANALYSIS:SPLIT_FASTQ (DemoAtacCombinatorial_S1-AAAGAA)                                [100%] 2 of 2 ✔
[1e/674df2] process > ATAC_ANALYSIS:CUTADAPT_HEADCROP (DemoAtacCombinatorial_S1-AAAGAA)                          [100%] 2 of 2 ✔
[93/b3e6f9] process > ATAC_ANALYSIS:BWA_ALIGNMENT (DemoAtacCombinatorial_S1-AAAGAA)                              [100%] 2 of 2 ✔
[2d/69b034] process > ATAC_ANALYSIS:MARK_DUPLICATES (DemoAtacCombinatorial_S1-TTTGGG)                            [100%] 2 of 2 ✔
[5c/058a63] process > ATAC_ANALYSIS:SPLIT_BAM (DemoAtacCombinatorial_S1-TTTGGG)                                  [100%] 2 of 2 ✔
[6a/372623] process > ATAC_ANALYSIS:ASSEMBLE_FRAGMENTS (DemoAtacCombinatorial_S1-TTTGGG, mouse.10)               [100%] 8 of 8 ✔
[1b/2a0edc] process > ATAC_ANALYSIS:ANNOTATE_FRAGMENTS (DemoAtacCombinatorial_S1-TTTGGG, mouse.10)               [100%] 8 of 8 ✔
[fa/f8e5ad] process > ATAC_ANALYSIS:DETERMINE_BARCODE_ALLOWLIST (DemoAtacCombinatorial_S1-TTTGGG)                [100%] 2 of 2 ✔
[2d/4f2f2b] process > ATAC_ANALYSIS:COMPUTE_DECONVOLUTION_STAT_CHR (DemoAtacCombinatorial_S1-TTTGGG, mouse.10) [100%] 8 of 8 ✔
[9a/2af17a] process > ATAC_ANALYSIS:DETERMINE_BARCODE_MERGES (DemoAtacCombinatorial_S1)                          [100%] 1 of 1 ✔
[df/7edf9d] process > ATAC_ANALYSIS:REANNOTATE_FRAGMENTS (DemoAtacCombinatorial_S1-AAAGAA, mouse.10)             [100%] 8 of 8 ✔
[89/120b90] process > ATAC_ANALYSIS:REANNOTATE_BAM (DemoAtacCombinatorial_S1-AAAGAA, mouse.10)                   [100%] 8 of 8 ✔
[78/155ef9] process > ATAC_ANALYSIS:FINAL_BAM_MERGE (DemoAtacCombinatorial_S1-TTTGGG)                            [100%] 2 of 2 ✔
[21/f6afcc] process > ATAC_ANALYSIS:MERGE_REANN_READ_COUNTS (DemoAtacCombinatorial_S1-TTTGGG)                    [100%] 2 of 2 ✔
[24/46c918] process > ATAC_ANALYSIS:FINAL_FRAG_MERGE (DemoAtacCombinatorial_S1-TTTGGG)                           [100%] 2 of 2 ✔
[15/df749e] process > ATAC_ANALYSIS:ASSEMBLE_BASIC_QC (DemoAtacCombinatorial_S1-TTTGGG)                          [100%] 2 of 2 ✔
[90/756c5c] process > ATAC_ANALYSIS:FINAL_QC_SE (DemoAtacCombinatorial_S1-TTTGGG)                                [100%] 2 of 2 ✔
[2d/89b1d4] process > ATAC_ANALYSIS:CALCULATE_BEADS_PER_DROP (DemoAtacCombinatorial_S1-TTTGGG)                   [100%] 2 of 2 ✔
[8b/d1fd55] process > ATAC_ANALYSIS:COMPILE_ALIGNMENTS (SuperloadedSample)                                       [100%] 1 of 1 ✔
[3d/e6eb53] process > ATAC_ANALYSIS:COMPILE_FRAGMENTS (SuperloadedSample)                                        [100%] 1 of 1 ✔
[c1/0e6474] process > ATAC_ANALYSIS:COMPILE_QC_STATS (SuperloadedSample)                                         [100%] 1 of 1 ✔
[97/fedc1c] process > ATAC_ANALYSIS:BEAD_FILT_SUMMARY (SuperloadedSample)                                        [100%] 1 of 1 ✔
[63/c06e3d] process > ATAC_ANALYSIS:SUMMARIZE_ALIGNMENTS (SuperloadedSample)                                     [100%] 1 of 1 ✔
[44/087f75] process > ATAC_ANALYSIS:COMPUTE_TSS_MATRIX (SuperloadedSample)                                       [100%] 1 of 1 ✔
[bc/07d63d] process > ATAC_ANALYSIS:CALL_PEAKS (SuperloadedSample)                                               [100%] 1 of 1 ✔
[26/98ad59] process > ATAC_ANALYSIS:CLEAN_PEAKS (SuperloadedSample)                                              [100%] 1 of 1 ✔
[90/416511] process > ATAC_ANALYSIS:FRACTION_OF_READS_IN_PEAKS (SuperloadedSample)                               [100%] 1 of 1 ✔
[bf/3c7049] process > ATAC_ANALYSIS:FRACTION_OF_READS_IN_TSS (SuperloadedSample)                                 [100%] 1 of 1 ✔
[aa/82d2bc] process > ATAC_ANALYSIS:CALCULATE_INSERT_SIZE_METRICS (SuperloadedSample)                            [100%] 1 of 1 ✔
[95/469541] process > ATAC_ANALYSIS:SUMMARIZE_MIXED_SPECIES (SuperloadedSample)                                  [100%] 1 of 1 ✔
[3e/d42c9f] process > ATAC_ANALYSIS:MAKE_COUNT_MATRIX (SuperloadedSample)                                        [100%] 1 of 1 ✔
[28/8854d6] process > ATAC_ANALYSIS:SEQUENCE_SATURATION (SuperloadedSample)                                      [100%] 1 of 1 ✔
[c4/dccf13] process > ATAC_ANALYSIS:ARCHR (SuperloadedSample)                                                    [100%] 1 of 1 ✔
[fc/4e9677] process > ATAC_ANALYSIS:TSS_ENRICHMENT (SuperloadedSample)                                           [100%] 1 of 1 ✔
[2a/bc5cc6] process > ATAC_ANALYSIS:AGGREGATE_METRICS                                                            [100%] 1 of 1 ✔
[5f/afd8e9] process > ATAC_ANALYSIS:BUILD_REPORT_CONTENTS                                                        [100%] 1 of 1 ✔
[d6/0d1100] process > ATAC_ANALYSIS:GENERATE_REPORTS (1)                                                         [100%] 1 of 1 ✔
Completed at: 09-Jun-2022 16:02:06
Duration    : 5m 20s
CPU hours   : 0.6
Succeeded   : 92
```

When the Analysis Workflow run is finished, the output directory should contain the files shown below. The main deliverables are presented under SuperloadedSample, a name given to the grouped combinatorial indexes in the superloading use case.

▶ To view the file list, run the `tree` command.



Analysis Workflow Example: Combinatorial ATAC-Seq Multiplexing

This example uses a mixed-species experiment in which the user has already done the following:

- Generated sequencing data using the SureCell ATAC-Seq Library Prep Kit from Bio-Rad.
- Used a set of indexed transposases for nuclear transposition.
- Conducted a multiplexing experiment, where the transposition indexes used on different samples are treated as distinct TIs by the analysis.
- Prepared a genome reference from Reference Workflow: Mixed Species on page 37 and saved it to the **/home/ubuntu/mixed-reference** directory.

A multiplexing ATAC-seq superloading analysis requires additional configuration. The preset default values in the Omnition configuration assume that you have conducted an experiment using all 96 transposition indexes, as in Lareau. The transposition index sequence makes up the first six bases following the bead barcode sequence in the first read of each read pair.

Note: This example uses a subset of the Lareau transposition indexes, which are defined in the configuration file.

In multiplexing scenarios, a map of transposition index to FASTQ file to sample name is also required. Omnition uses this information to split and merge data during analysis, and map it to the appropriate sample.

The reads in this demonstration are paired-end reads that were not lane split during demultiplexing. If lane split reads are provided, Omnition merges them during analysis.

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Prerequisite: Create and change to the combinatorial-atac-reads directory

▶ Use the following commands to create and change to the **combinatorial-atac-reads** directory.

```
mkdir /home/ubuntu/combinatorial-atac-reads

cd /home/ubuntu/combinatorial-atac-reads
```

Step 1: Download the FASTQ files to the combinatorial-atac-reads directory

Important: The download commands in this step apply to the demo files only. When using your own files, retrieve them from the applicable storage location.

▶ From the **/home/ubuntu/combinatorial-atac-reads** directory, use the following commands to download the demo FASTQ files.

```
cp ~/.nextflow/assets/BioRadOpenSource/omnition/test/data/atac/combinatorial/
*.fastq.gz ./
```

Step 2: Create the FASTQ-TI Configuration CSV

The FASTQ-TI configuration CSV file maps a sample name to a FASTQ file, and a TI within that file. The configuration CSV contains a header and two sample maps to the FASTQ file. For information on the CSV file, see FASTQ-TI Configuration (Multiplexing) on page 22.

In the following example, one FASTQ file (**DemoAtacCombinatorial_S1**) contains two TIs, **AAAGAA** in **ti1** and **TTTGGG** in **ti2** (shown in the workflow configuration file), which correspond to two samples (SampleA and SampleB).

A formatted version of the file is available in the Omnition repository in GitHub. For information, see GitHub on page 69.

To create and save the CSV configuration file

1. Create the file using the following format:

```
sample,fastq,ti
SampleA,DemoAtacCombinatorial_S1,ti1
SampleB,DemoAtacCombinatorial_S1,ti2
```

2. Save the file as **sample-map.csv** in the **/home/ubuntu/combinatorial-atac-reads** directory.

Important: The directories and paths in your environment can be different than those specified in this example. For information, see Read Me First on page 13.

Step 3: Create the workflow configuration and indexing file

The configuration file for the multiplexing workflow is identical to the superloading workflow, except for an additional parameter at the end, **barcodedTn5config**, which points to the path to the FASTQ-TI configuration CSV file.

```
atac:
    workflow: "analysis"
    input: "/home/ubuntu/combinatorial-atac-reads"
    reference:
        directory: "/home/ubuntu/mixed-genome"
        fasta:
            species1: "/home/ubuntu/mixed-genome/human.fa.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.fa.gz"
        gtf:
            species1: "/home/ubuntu/mixed-genome/human.gtf.gz"
            species2: "/home/ubuntu/mixed-genome/mouse.gtf.gz"
        blocklist: "/home/ubuntu/mixed-genome/filtered.blocklist.bed"
    mixed: true
    barcodedTn5: true
    ti:
        ti1: "AAAGAA"
        ti2: "TTTGGG"
    tiread: "r1"
    barcodedTn5Config: "/home/ubuntu/combinatorial-atac-reads/sample-map.csv"
```

To create and save the file

1. Complete the following substeps to create the configuration file, as shown in the above example.
    a. Enter the assay type parameter (**atac**) and the workflow parameter (**analysis**).
    b. Below the **workflow** parameter, enter the **input** parameter, which is the path to the FASTQ files for the analysis run.
    c. Below the **input** parameter, enter the **reference** parameter.

d. Below the **reference** parameter, enter the **directory** parameter, with the path to the reference directory, and then enter the **fasta** and **gtf** parameters, with paths to the file sets downloaded from ENSEMBL.

   Note: For mixed species workflows, you must specify a file for each species under **fasta** and **gtf**.

e. To add a **blocklist** parameter, enter **blocklist** and then enter the path to the blocklist file.

f. To indicate a mixed species experiment, set the **mixed** parameter to **true**.

g. To indicate a combinatorial indexing workflow, set the **barcodedTn5** parameter to **true**.

h. Below **barcodedTn5**, enter the **ti** parameter, and then enter the specific TIs.

   Note: This example defines the TIs as part of a list and tells Omnition that the TI is the first six bases following the barcode in the first read pair.

i. Enter the **tiread** parameter and specify the read.

j. Enter the **barcodedTn5config** parameter, and enter the path to the FASTQ-TI configuration CSV file.

2. Save the file as **analysis-multiplex.yaml** in the **/home/ubuntu/combinatorial-atac-reads** directory.

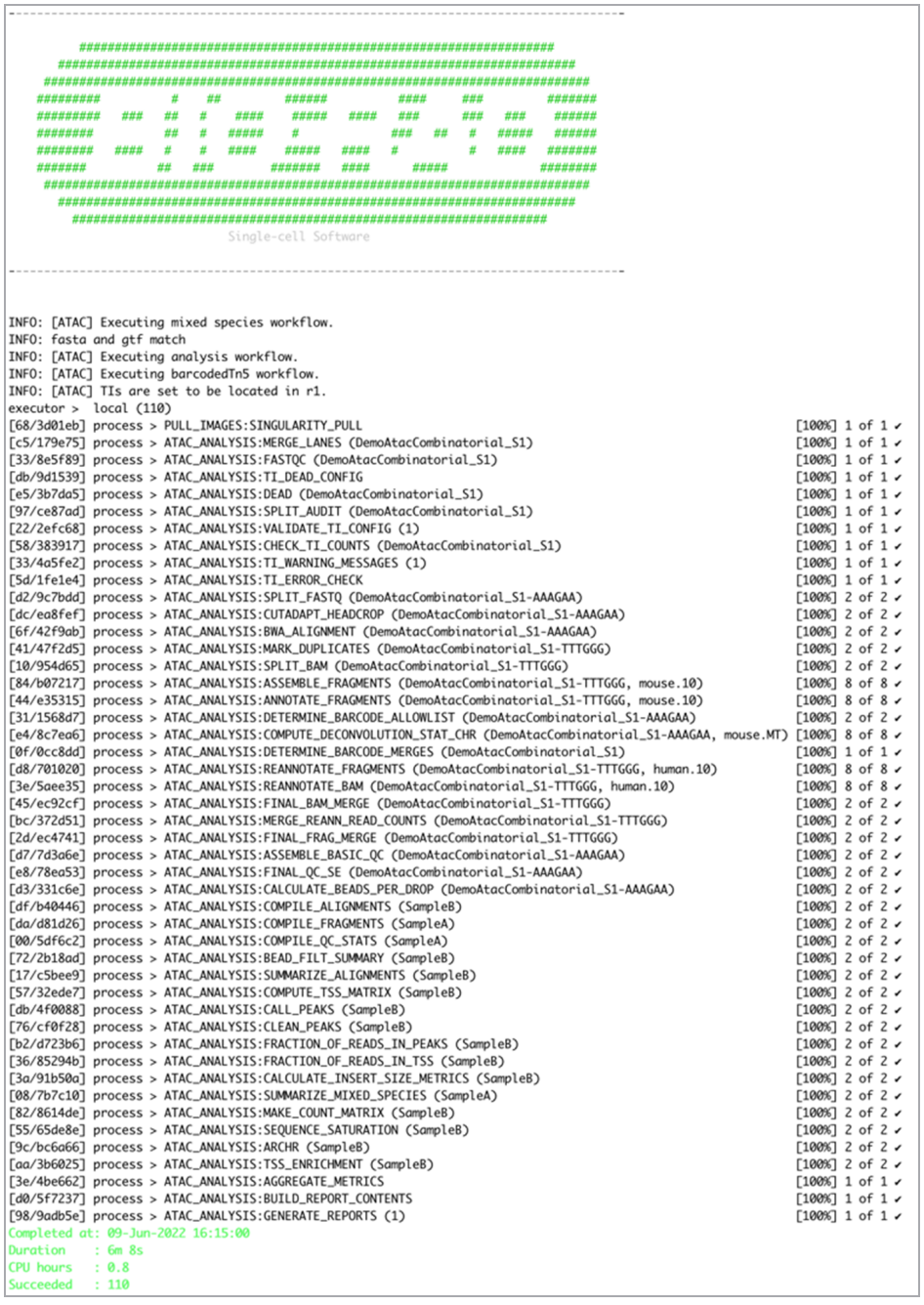
Step 4: Run the Analysis Workflow

▶ Run the following command in Nextflow to run the Analysis Workflow.

```
nextflow run BioRadOpenSource/omnition \
  -params-file analysis-multiplex.yaml \
  -profile standard
```

Step 5: Verify the outputs

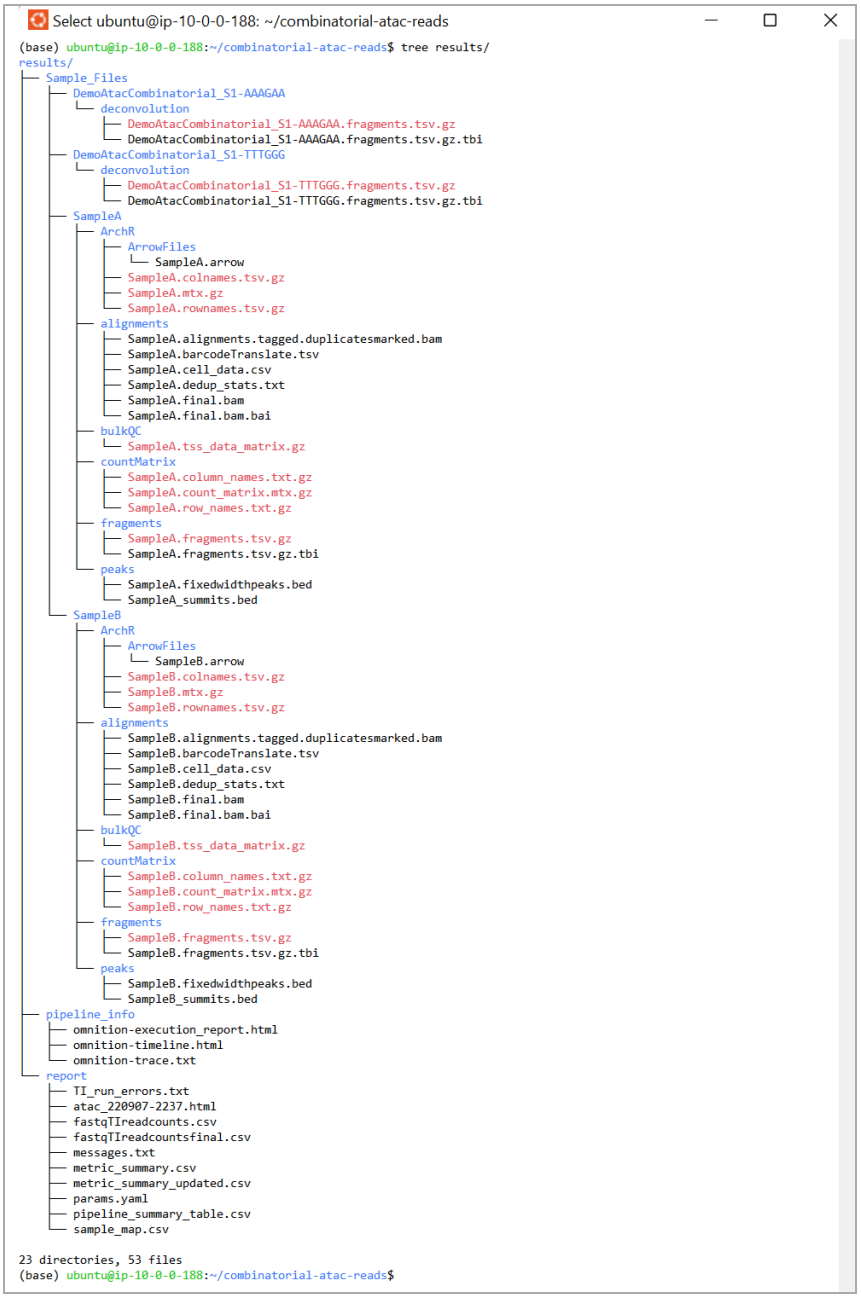
As the Analysis Workflow runs, a progress report appears on the screen.



```
--------------------------------------------------------------------------------

        ############################################################
      ################################################################
    ####################################################################
   #########         #    ##        ######         ####     ###        #######
   #########   ###   ##   #    ####    #####   ####   ###      ###   ###    ######
   ########          ##   #   #####    #           ###   ##   #   #####   ######
   ########   ####   #    #   ####    #####   ####   #         #   ####   #######
   #######          ##   ###        #######   ####      #####            ########
    ##################################################################
      ##############################################################
        ##########################################################
                              Single-cell Software

--------------------------------------------------------------------------------


INFO: [ATAC] Executing mixed species workflow.
INFO: fasta and gtf match
INFO: [ATAC] Executing analysis workflow.
INFO: [ATAC] Executing barcodedTn5 workflow.
INFO: [ATAC] TIs are set to be located in r1.
executor >  local (110)
[68/3d01eb] process > PULL_IMAGES:SINGULARITY_PULL                                                    [100%] 1 of 1 ✔
[c5/179e75] process > ATAC_ANALYSIS:MERGE_LANES (DemoAtacCombinatorial_S1)                            [100%] 1 of 1 ✔
[33/8e5f89] process > ATAC_ANALYSIS:FASTQC (DemoAtacCombinatorial_S1)                                 [100%] 1 of 1 ✔
[db/9d1539] process > ATAC_ANALYSIS:TI_DEAD_CONFIG                                                    [100%] 1 of 1 ✔
[e5/3b7da5] process > ATAC_ANALYSIS:DEAD (DemoAtacCombinatorial_S1)                                   [100%] 1 of 1 ✔
[97/ce87ad] process > ATAC_ANALYSIS:SPLIT_AUDIT (DemoAtacCombinatorial_S1)                            [100%] 1 of 1 ✔
[22/2efc68] process > ATAC_ANALYSIS:VALIDATE_TI_CONFIG (1)                                            [100%] 1 of 1 ✔
[58/383917] process > ATAC_ANALYSIS:CHECK_TI_COUNTS (DemoAtacCombinatorial_S1)                        [100%] 1 of 1 ✔
[33/4a5fe2] process > ATAC_ANALYSIS:TI_WARNING_MESSAGES (1)                                           [100%] 1 of 1 ✔
[5d/1fe1e4] process > ATAC_ANALYSIS:TI_ERROR_CHECK                                                    [100%] 1 of 1 ✔
[d2/9c7bdd] process > ATAC_ANALYSIS:SPLIT_FASTQ (DemoAtacCombinatorial_S1-AAAGAA)                     [100%] 2 of 2 ✔
[dc/ea8fef] process > ATAC_ANALYSIS:CUTADAPT_HEADCROP (DemoAtacCombinatorial_S1-AAAGAA)               [100%] 2 of 2 ✔
[6f/42f9ab] process > ATAC_ANALYSIS:BWA_ALIGNMENT (DemoAtacCombinatorial_S1-AAAGAA)                   [100%] 2 of 2 ✔
[41/47f2d5] process > ATAC_ANALYSIS:MARK_DUPLICATES (DemoAtacCombinatorial_S1-TTTGGG)                 [100%] 2 of 2 ✔
[10/954d65] process > ATAC_ANALYSIS:SPLIT_BAM (DemoAtacCombinatorial_S1-TTTGGG)                       [100%] 2 of 2 ✔
[84/b07217] process > ATAC_ANALYSIS:ASSEMBLE_FRAGMENTS (DemoAtacCombinatorial_S1-TTTGGG, mouse.10)    [100%] 8 of 8 ✔
[44/e35315] process > ATAC_ANALYSIS:ANNOTATE_FRAGMENTS (DemoAtacCombinatorial_S1-TTTGGG, mouse.10)    [100%] 8 of 8 ✔
[31/1568d7] process > ATAC_ANALYSIS:DETERMINE_BARCODE_ALLOWLIST (DemoAtacCombinatorial_S1-AAAGAA)     [100%] 2 of 2 ✔
[e4/8c7ea6] process > ATAC_ANALYSIS:COMPUTE_DECONVOLUTION_STAT_CHR (DemoAtacCombinatorial_S1-AAAGAA, mouse.MT) [100%] 8 of 8 ✔
[0f/0cc8dd] process > ATAC_ANALYSIS:DETERMINE_BARCODE_MERGES (DemoAtacCombinatorial_S1)              [100%] 1 of 1 ✔
[d8/701020] process > ATAC_ANALYSIS:REANNOTATE_FRAGMENTS (DemoAtacCombinatorial_S1-TTTGGG, human.10)  [100%] 8 of 8 ✔
[3e/5aee35] process > ATAC_ANALYSIS:REANNOTATE_BAM (DemoAtacCombinatorial_S1-TTTGGG, human.10)        [100%] 8 of 8 ✔
[45/ec92cf] process > ATAC_ANALYSIS:FINAL_BAM_MERGE (DemoAtacCombinatorial_S1-TTTGGG)                 [100%] 2 of 2 ✔
[bc/372d51] process > ATAC_ANALYSIS:MERGE_REANN_READ_COUNTS (DemoAtacCombinatorial_S1-TTTGGG)         [100%] 2 of 2 ✔
[2d/ec4741] process > ATAC_ANALYSIS:FINAL_FRAG_MERGE (DemoAtacCombinatorial_S1-TTTGGG)               [100%] 2 of 2 ✔
[d7/7d3a6e] process > ATAC_ANALYSIS:ASSEMBLE_BASIC_QC (DemoAtacCombinatorial_S1-AAAGAA)              [100%] 2 of 2 ✔
[e8/78ea53] process > ATAC_ANALYSIS:FINAL_QC_SE (DemoAtacCombinatorial_S1-AAAGAA)                     [100%] 2 of 2 ✔
[d3/331c6e] process > ATAC_ANALYSIS:CALCULATE_BEADS_PER_DROP (DemoAtacCombinatorial_S1-AAAGAA)        [100%] 2 of 2 ✔
[df/b40446] process > ATAC_ANALYSIS:COMPILE_ALIGNMENTS (SampleB)                                      [100%] 2 of 2 ✔
[da/d81d26] process > ATAC_ANALYSIS:COMPILE_FRAGMENTS (SampleA)                                       [100%] 2 of 2 ✔
[00/5df6c2] process > ATAC_ANALYSIS:COMPILE_QC_STATS (SampleA)                                        [100%] 2 of 2 ✔
[72/2b18ad] process > ATAC_ANALYSIS:BEAD_FILT_SUMMARY (SampleB)                                       [100%] 2 of 2 ✔
[17/c5bee9] process > ATAC_ANALYSIS:SUMMARIZE_ALIGNMENTS (SampleB)                                    [100%] 2 of 2 ✔
[57/32ede7] process > ATAC_ANALYSIS:COMPUTE_TSS_MATRIX (SampleB)                                      [100%] 2 of 2 ✔
[db/4f0088] process > ATAC_ANALYSIS:CALL_PEAKS (SampleB)                                              [100%] 2 of 2 ✔
[76/cf0f28] process > ATAC_ANALYSIS:CLEAN_PEAKS (SampleB)                                             [100%] 2 of 2 ✔
[b2/d723b6] process > ATAC_ANALYSIS:FRACTION_OF_READS_IN_PEAKS (SampleB)                              [100%] 2 of 2 ✔
[36/85294b] process > ATAC_ANALYSIS:FRACTION_OF_READS_IN_TSS (SampleB)                                [100%] 2 of 2 ✔
[3a/91b50a] process > ATAC_ANALYSIS:CALCULATE_INSERT_SIZE_METRICS (SampleB)                           [100%] 2 of 2 ✔
[08/7b7c10] process > ATAC_ANALYSIS:SUMMARIZE_MIXED_SPECIES (SampleA)                                 [100%] 2 of 2 ✔
[82/8614de] process > ATAC_ANALYSIS:MAKE_COUNT_MATRIX (SampleB)                                       [100%] 2 of 2 ✔
[55/65de8e] process > ATAC_ANALYSIS:SEQUENCE_SATURATION (SampleB)                                     [100%] 2 of 2 ✔
[9c/bc6a66] process > ATAC_ANALYSIS:ARCHR (SampleB)                                                   [100%] 2 of 2 ✔
[aa/3b6025] process > ATAC_ANALYSIS:TSS_ENRICHMENT (SampleB)                                          [100%] 2 of 2 ✔
[3e/4be662] process > ATAC_ANALYSIS:AGGREGATE_METRICS                                                 [100%] 1 of 1 ✔
[d0/5f7237] process > ATAC_ANALYSIS:BUILD_REPORT_CONTENTS                                             [100%] 1 of 1 ✔
[98/9adb5e] process > ATAC_ANALYSIS:GENERATE_REPORTS (1)                                              [100%] 1 of 1 ✔
Completed at: 09-Jun-2022 16:15:00
Duration    : 6m 8s
CPU hours   : 0.8
Succeeded   : 110
```

When the Analysis Workflow run is finished, the output directory should contain the files shown below.

▶ To view the file list, run the `tree` command.



Pipeline Outputs

This section describes the outputs produced by Omnition Analysis Software.

Table 7. Pipeline Outputs

Output	Description
HTML report	*A single HTML report with quality control metrics* If Omnition performed a multiple sample batch analysis, you can select individual samples from the dropdown list to view the populated metrics and results.
BAM files	*A BAM file for each sample in a pipeline run* The BAM file is named with the sample name and has the extension **.final.bam**. It contains the primary alignment for each read input to the experiment, deduplicated at the cell level where PCR duplicates within the same partition are represented by a single representative fragment.
BAM tags	*A BAM tag reports the primary alignment for each representative fragment from each partition* BAM tags are annotated on the records present in the BAM files. Each alignment has the MC, MD, NM, AS, and XS generated by BWA-MEM. The PG tag is annotated by Picard MarkDuplicates. Omnition uses the XB tag for the bead barcode and the DB tag for the cell barcode.
Fragments file	*A BED-like file with the following values in each column:* ■ Reference sequence chromosome or contig name ■ Position of fragment alignment start site on chromosome or contig ■ Position of fragment alignment end site on chromosome or contig ■ Cell barcode for this fragment ■ Number of fragments with these alignment coordinates
Reads-in-peaks count matrix	*Peaks are called on pseudo-bulk alignments (the alignments with cell barcodes are ignored)* For each filtered cell, every alignment is intersected with the pseudo-bulk peaks and a per-cell count is registered for each intersection. The reads-in-peaks counts matrix is formatted in a sparse matrix format, where columns represent the row index, the column index, and the value at that coordinate. The row and column names are the cell barcodes and peak names, respectively, and stored in the accompanying **.row_names.txt.gz** and **.column_names.txt.gz** files.

This page intentionally left blank.

Appendix A Performance Benchmarks

This section provides benchmarks for Omnition Analysis Software when running on an Amazon Web Services (AWS) EC2 instance. Analysis was performed on an AWS R5ad.4xlarge system with 16 AMD EPYC CPUs and 128 GB of RAM. The default resource allocation settings for Omnition are set for a system of this size. To adjust resource allocations for larger or smaller systems, see Adjusting Resource Allocations on page 67.

Reference Workflow Performance

Table 8 identifies performance benchmarks for the ATAC-seq Reference Workflow when building a reference from the primary assembly of GRCh38.

Table 8. Performance Benchmarks for the Reference Workflow

Workflow	Process Name	Time (mins)	% CPU	RAM (MB)
ATAC_REFERENCE	BWA_INDEX	69.7	98.3	4300
ATAC_REFERENCE	ARCHR_REFERENCE	7.6	92.0	6900
ATAC_REFERENCE	GUNZIP_FASTA	0.6	96.5	61.6
ATAC_REFERENCE	FILTER_REFERENCES	0.5	96.0	63.9
ATAC_REFERENCE	GENERATE_GENOME_SIZES	0.4	113.4	11.5
ATAC_REFERENCE	FILTER_BLOCKLISTS	0.4	95.0	64.5
ATAC_REFERENCE	GENERATE_TSS_WINDOWS	0.4	141.5	90.6
ATAC_REFERENCE	GUNZIP_GTF	0.3	93.3	59.7
ATAC_REFERENCE	GENERATE_EMPTY_ BLOCKLIST	0.2	89.4	56.9

Datasets

As shown in Table 9, benchmarks were calculated for three production-sized datasets of cells (PBMCs), as follows:

- Datasets a) and b) were sequenced at approximately 42,000 reads per cell.
- Dataset c) was sequenced at approximately 118,000 reads per cell, and is included to illustrate the pipeline performance in processing higher volume data.

Table 9. Pre-processing data characteristics

	Sample Name	Total Reads	Total Cells	Reads/ Cells
a)	N707-ATACExp55-Sample7_S1	114,310,431	2,741	~42K
b)	N710-ATACExp55-Sample8_S5	117,075,800	2,649	~44K
c)	Bio-Rad-ATACseq4_S4	520,568,170	4,963	~118K
	Entire batch	751,954,401	10,353	~72K

As shown in Table 10, the analysis of each sample was repeated at 25%, 50%, and 75% of the input to set expectations for performance at different sequencing depths.

Table 10. Processing time performance

Sample Name	100%	75%	50%	25%
Entire batch	2d 3h 38m 33s	21h 48m 45s	15h 56m 28s	9h 55m 37s

Due to the parallel processing performed by Nextflow, the summed compute times for each process for the individual samples within the batch are greater than the overall batch processing wall-clock time. Therefore, only the batch processing (wall-clock) time is reported. This is the amount of time the user would be expected to wait for a pipeline run of this size to complete on a system with similar specifications.

Appendix B Troubleshooting

This section contains answers to frequently asked questions regarding Omnition Analysis Software.

To contact Bio-Rad Technical Support, send an email to support@bio-rad.com, and include the workflow configuration file and the Nextflow log file, which is stored in the directory from which you launched the Omnition workflow pipeline. Additional Tech Support contact information is available at in the front of this guide.

Frequently Asked Questions

Do I need to clone the Omnition GitHub repository in order to use the software?

No. When you run Omnition, Nextflow clones the repository to your home directory path by default:

~/.nextflow/assets/BioRadOpenSource/omnition

How does Omnition perform bead merging?

Omnition uses the approach that was published in Lareau. The source code for bead-based ATAC processing (BAP) was modified and integrated into Omnition. The core logic for bead merging remains the same. For information on BAP, see https://github.com/calebrlareau/bap.

How does Omnition differ from the Bio-Rad ATAC-Seq Analysis Toolkit?

Omnition is a replacement for the Bio-Rad ATAC-Seq Analysis Toolkit. The software includes significant performance and usability improvements, including the ability to process samples in batches, streamlined multi-sample batch reports, and workflow organization and management by Nextflow.

Important: In building Omnition, new algorithms were developed that might result in slightly different quality control metrics, as described below:

- New barcode parsing and correction algorithm with greater stringency might result in a lower percentage of reads with valid barcodes in some experiments.
- New TSS enrichment scoring method and use of ENSEMBL annotations might result in slightly lower TSS enrichment scores in some experiments.
- Changes in the method for calculating duplicate reads might result in lower duplicate rates in some experiments.

For more information and a complete list of frequently asked questions, see the information at https://www.bio-rad/omnition.

This page intentionally left blank.

Appendix C Installation

Setting Up the Environment Applications

This section describes the sequential tasks to install the applications necessary to run Omnition Analysis Software, and assumes the following:

- The installer is a system administrator with root-level permissions on the system.
- The installer is starting with a clean environment.

All required installations are documented with demonstration commands for an Ubuntu 21.10 system using the apt-get package management tool.

Note: If you are using a shared computing environment (for example, an HPC cluster), some or all of these applications might already be installed with other commonly used software. If so, skip the applicable installation sections.

Step 1: Install Singularity or Docker

You can install both applications, but only one is required. Complete the steps in the following subsections to install the preferred application.

Singularity

Use the commands in the following steps to complete the Singularity installation. The code blocks in each section are duplicated from the Singularity documentation. For information, click the following hyperlink:

https://docs.sylabs.io/guides/3.0/user-guide/installation.html#install-dependencies

To set up Singularity

1. Install the specified dependencies using the apt-get command line interface.

```
sudo apt-get update && sudo apt-get install -y \
  build-essential \
  libssl-dev \
  uuid-dev \
  libgpgme11-dev \
  squashfs-tools \
  libseccomp-dev \
  pkg-config
```

2. Install Go.

```
export VERSION=1.11 OS=linux ARCH=amd64 && \
  wget https://dl.google.com/go/go$VERSION.$OS-$ARCH.tar.gz && \
  sudo tar -C /usr/local -xzvf go$VERSION.$OS-$ARCH.tar.gz && \
  rm go$VERSION.$OS-$ARCH.tar.gz
```

3. Set up the Go environment.

```
echo 'export GOPATH=${HOME}/go' >> ~/.bashrc && \
  echo 'export PATH=/usr/local/go/bin:${PATH}:${GOPATH}/bin' >> ~/.bashrc && \
  source ~/.bashrc
```

4. Install Singularity.

   Important: Ensure the release version is correct before runnning the commands.

```
export VERSION=3.6.4 && \
  mkdir -p $GOPATH/src/github.com/sylabs && \
  cd $GOPATH/src/github.com/sylabs && \
  wget https://github.com/sylabs/singularity/releases/download/
  v${VERSION}/singularity-${VERSION}.tar.gz && tar -xzf singularity-
  ${VERSION}.tar.gz && \
  cd ./singularity && \
  ./mconfig
```

5. Compile Singularity.

```
./mconfig && \
  make -C builddir && \
  sudo make -C builddir install
```

6. Verify the Singularaity installation.

```
singularity pull docker://godlovedc/lolcow && \
singularity exec lolcow_latest.sif cowsay "hello world"
```

Docker

Use the commands specified below to complete the Docker installation.

Important: The commands are taken directly from the Docker documentation. For information, click the following hyperlink:

https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository

To set up Docker

1. Install the specified dependencies using the apt-get command line interface.

```
sudo apt-get update

sudo apt-get install \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
```

   When prompted to continue, enter Y.

2. Add the GPG key.

```
sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg -- \
  dearmor -o /etc/apt/keyrings/docker.gpg
```

3. Set up the repository.

```
echo \
  "deb [arch=$(dpkg --print-architecture) \
signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4. Install the Docker Engine.

```
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io \
  docker-compose-plugin
```

5. Test the Docker Engine.

```
sudo docker run hello-world
```

Note: Steps 6 and 7 instruct Linux to allow non-privileged users to run Docker commands:

6. Create the Docker group

```
sudo groupadd docker
```

7. Add the current user to the Docker group.

```
sudo usermod -aG docker $USER
```

Step 2: Install Miniconda and Nextflow

Use the commands specified below to complete the Miniconda, Nextflow, and Java installations.

Note: Java is installed as part of the Nexflow installation.

Bio-Rad recommends using the Conda package system to manage your Nextflow environment.

1. Install Miniconda.

   Important: Miniconda is a simplified version of Conda and the commands are taken directly from the Conda documentation. Miniconda is used to complete the remaining installations. `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh bash Miniconda3-latest-Linux-x86_64.sh`

2. Create the Nextflow environment

   ```
   conda create --name nextflow
   ```

3. Activate the Nextflow environment.

   ```
   conda activate nextflow
   ```

4. Install Nextflow.

   ```
   conda install -c bioconda nextflow=21.0.4
   ```

   Note: If applicable, replace 21.0.4 with the later version you want to install. See Software Requirements on page 10 for the recommended Nextflow versions.

5. Test the Nextflow installation.

   ```
   nextflow -version
   ```

Note: If Nextflow is installed in its own Conda environment, you must activate that environment before running Omnition Analysis Software.

This page intentionally left blank.

Default Resource Allocations

Omnition Analysis Software is composed of a series of processes that connect to form a Nextflow pipeline. Each process, when it is launched by the Nextflow scheduler, is allocated a portion of the host system CPUs and memory (RAM).

Note: Processes are launched in parallel as system capacity permits. If a process fails due to insufficient resources, it is retried with the resource allocation doubled. If the process fails a second time, no further retries are carried out.

The tables below provide CPU and RAM default resource allocations, by label according to size, to optimize Omnition process performance on a system with 16 CPUs and 64 GB RAM. If you are running combinatorial ATAC-seq workflows, Bio-Rad recommends a minimum of 16 CPUs and 128 GB RAM.

Note: For information on resource labels for each process, see the Omnition README file in the GitHub repository.

Table 11. CPU resource allocation

Label	CPUs
cpu_xsmall	1
cpu_small	2
cpu_medium	4
cpu_large	8
cpu_xlarge	16

Table 12. Memory resource allocation

Label	RAM (GB)
memory_xxsmall	0.9375
memory_xsmall	3.7500
memory_small	7.5
memory_medium	15.0
memory_large	30.0
memory_xlarge	60.0

Adjusting Resource Allocations

To change the memory allocation for processes with the memory_xlarge label, create a configuration file in the format shown below, and save it as **resources.config** in the directory where the workflow configuration file is stored.

```
process{
    withLabel: memory_xlarge {
        memory = 128.GB
        time = 24.h
    }
}
```

Note: A formatted version of this file is available in the Omnition GitHub repository.

After changing the memory allocation, save the file, and then use the following commands to pass it to Omnition when running the software.

```
nextflow run BioRadOpenSource/omnition \
  -params-file analysis.yaml \
  -profile standard \
  -c resources.config
```

The allocations in **resources.config** override the default settings.

About Nextflow and Omnition Analysis Software

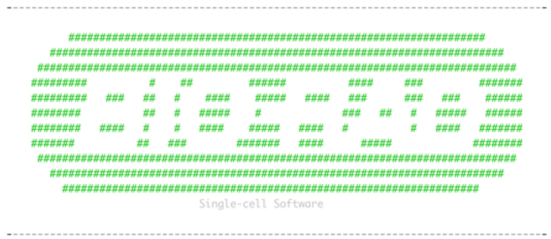
Omnition Analysis Software is a Nextflow pipeline, and the following sections summarize the Nextflow functionality that is supported in Omnition. For information regarding Nextflow, see the documentation at the following link:

https://www.nextflow.io/docs/latest/index.html

Nextflow Work Directories

Nextflow stores temporary workflow data in a directory named **work**, and locates the directory in the same directory location from which you launch the Omnition pipeline.

Nextflow runs each Omnition process in a subdirectory within the **work** directory. The path of the subdirectory appears in the run progress report, to the left of the process name, as shown in the following example.




```
INFO: [ATAC] Executing mixed species workflow.
INFO: fasta and gtf match
INFO: [ATAC] Executing analysis workflow.
INFO: [ATAC] Executing barcodedTn5 workflow.
INFO: [ATAC] TIs are set to be located in r1.
executor >  local (110)
[68/3d01eb] process > PULL_IMAGES:SINGULARITY_PULL                                [100%] 1 of 1 ✔
[c5/179e75] process > ATAC_ANALYSIS:MERGE_LANES (DemoAtacCombinatorial_S1)        [100%] 1 of 1 ✔
[33/8e5f89] process > ATAC_ANALYSIS:FASTQC (DemoAtacCombinatorial_S1)             [100%] 1 of 1 ✔
[db/9d1539] process > ATAC_ANALYSIS:TI_DEAD_CONFIG                               [100%] 1 of 1 ✔
[e5/3b7da5] process > ATAC_ANALYSIS:DEAD (DemoAtacCombinatorial_S1)              [100%] 1 of 1 ✔
[97/ce87ad] process > ATAC_ANALYSIS:SPLIT_AUDIT (DemoAtacCombinatorial_S1)       [100%] 1 of 1 ✔
[22/2efc68] process > ATAC_ANALYSIS:VALIDATE_TI_CONFIG (1)                       [100%] 1 of 1 ✔
[58/383917] process > ATAC_ANALYSIS:CHECK_TI_COUNTS (DemoAtacCombinatorial_S1)   [100%] 1 of 1 ✔
[33/4a5fe2] process > ATAC_ANALYSIS:TI_WARNING_MESSAGES (1)                      [100%] 1 of 1 ✔
[5d/1fe1e4] process > ATAC_ANALYSIS:TI_ERROR_CHECK                               [100%] 1 of 1 ✔
[d2/9c7bdd] process > ATAC_ANALYSIS:SPLIT_FASTQ (DemoAtacCombinatorial_S1-AAAGAA) [100%] 2 of 2 ✔
[dc/ea8fef] process > ATAC_ANALYSIS:CUTADAPT_HEADCROP (DemoAtacCombinatorial_S1-AAAGAA) [100%] 2 of 2 ✔
[6f/42f9ab] process > ATAC_ANALYSIS:BWA_ALIGNMENT (DemoAtacCombinatorial_S1-AAAGAA) [100%] 2 of 2 ✔
[41/47f2d5] process > ATAC_ANALYSIS:MARK_DUPLICATES (DemoAtacCombinatorial_S1-TTTGGG) [100%] 2 of 2 ✔
```

If necessary, you can inspect these work directories and subdirectories during troubleshooting.

Important: Because the **work** directory can require substantial disk space during processing, Bio-Rad recommends that you delete the **work** directory and subdirectories after the pipeline run is finished and you have verified the outputs.

Supported Nextflow Functionality

Omnition Analysis Software supports the following Nextflow functionality:

- When the **-resume** flag is present, along with the **.nextflow** and **work** directories, Nextflow resumes interrupted or failed pipeline runs from their stopping point.
- Nextflow produces the run report, timeline report, and trace report, and places them in the **pipeline_info** directory within the **output** directory.

Important: Bio-Rad has not tested Nextflow compatibility with Omnition Analysis Software beyond what is documented in this user guide. Bio-Rad does not support or assist with implementation or troubleshooting of functionality that is not explicitly named herein, or presented in the Omnition repository in GitHub.

GitHub

Omnition Analysis Software is distributed via GitHub, from the following repository:

http://github.com/BioRadOpenSource/omnition

Only released software is available from the repository, along with documentation that provides transparency regarding how the pipeline works and allows end users to modify the software to suit their needs.

To register issues you encounter when using the software, send an email to support@bio-rad.com. Responses might be delayed if you post the issue on GitHub.

Configuration Errors and Warnings

This section contains a complete list of configuration errors and warnings you might encounter when setting up Omnition Analysis Software.

- **The following errors appear if you did not use the required workflow run parameters.**
  - ERROR: [ATAC] The workflow parameter must be reference, analysis, or full.
  - ERROR: [ATAC] Must set the reference directory parameter.
- **The following errors appear if you did not meet the reference input file requirements.**
  - ERROR: [ATAC] The reference fasta parameter requires one file. If using two references, set mixed: true in the parameters file.
  - ERROR: [ATAC] The reference fasta parameter requires two files when mixed: true is set in the parameters file.
  - ERROR: [ATAC] The reference fasta parameter file does not exist: $value
  - ERROR: [ATAC] The reference fasta parameters must end in .fa, .fa.gz, .fna, .fna.gz, .fasta, or .fasta.gz.
  - ERROR: [ATAC] Must set the reference fasta parameter.
- **The following errors appear if you did not meet the multi-reference input file requirements.**
  - ERROR: [ATAC] The reference gtf parameter requires one file. If using two references, set mixed: true in the parameters file.
  - ERROR: [ATAC] The reference gtf parameter requires two files when mixed: true is set in the parameters file.
  - ERROR: [ATAC] The reference gtf parameter file does not exist: $value
  - ERROR: [ATAC] The reference gtf parameters must end in .gtf or .gtf.gz.
  - ERROR: [ATAC] Must set the reference gtf parameter.
  - ERROR: [ATAC] The contaminant directory parameter must be set if providing contaminants.
  - ERROR: [ATAC] The contaminant fasta parameter file does not exist: $value
  - ERROR: [ATAC] The contaminant fasta parameters must end in .fa, .fa.gz, .fna, .fna.gz, .fasta, or .fasta.gz.
  - ERROR: [ATAC] The mixed parameter must be Boolean (true or false).
- **The following errors appear if you did not meet input file setting requirements.**

- ERROR: [ATAC] The tierroroverride parameter must be Boolean (true or false).
- ERROR: [ATAC] The barcodedTn5 parameter must be Boolean (true or false).
- ERROR: [ATAC] The BarcodedTn5Config parameter file must be a csv file $value
- ERROR: [ATAC] The BarcodedTn5Config parameter file does not exist: $value
- ERROR: [ATAC] The tiread parameter cannot be used in conjunction with the i7asti parameter.
- ERROR: [ATAC] The tiread parameter must be r1, or r2.
- ERROR: [ATAC] The i7asti parameter must be Boolean (true or false).
- ERROR: [ATAC] The BarcodedTn5 parameter must be true in order to set i7asti parameter to true.
- ERROR: [ATAC] The config parameter file does not exist: $value
- ERROR: [ATAC] The input parameter directory does not exist: $value

The following errors appear if you did not meet blocklist file requirements.

- ERROR: [ATAC] The reference blocklist parameter accepts at most one file per reference genome. If using two references, set mixed: true in the parameters file.
- ERROR: [ATAC] The reference blocklist parameter accepts at most one file per reference genome.
- ERROR: [ATAC] The reference blocklist parameter file does not exist: $value
- ERROR: [ATAC] The reference blocklist parameters must end in blocklist.bed.

The following errors appear if you did not meet threshold requirements.

- ERROR: [ATAC] The tssWindowSize parameter must be an even integer.
- ERROR: [ATAC] The qualityThreshold parameter must be an integer.
- ERROR: [ATAC] The qualityThreshold parameter is not within a valid range.
- ERROR: [ATAC] The mergeMethod parameter must be one of both, r1, or r2.
- ERROR: [ATAC] The crosstalkthreshold parameter must be a float less than 1 and greater than 0.5.
- ERROR: [ATAC] The shadowthreshold parameter must be a float less than 1 and greater than 0.5.
- ERROR: [ATAC] The rounding parameter must be one of 0, 10, 100, or 1000.
- ERROR: [ATAC] The maxInsertSize parameter must be at least 100.

- **The following errors appear if you did not meet general workflow requirements.**
  - ERROR: [ATAC] No FASTQ files found in input directory. Check parameters and file name requirements.
  - ERROR: [ATAC] Cell $setting parameter must be provided as an integer: $value
  - ERROR: [ATAC] Barcode $setting parameter must be provided as an integer: $value
  - ERROR: [ATAC] The trim parameter must be provided as an integer: $value
  - ERROR: [ATAC] The mitoContig parameter must be a string.
  - ERROR: [ATAC] Must set $key parameter.
    - *This error appears when a parameter has not been set during configuration. **$key** is automatically replaced by the missing parameter.*
  - ERROR: [ATAC] FASTA and GTF file prefixes must match.
  - ERROR: [ATAC] No GTF files found in input directory. Check parameters and file name requirements.
  - ERROR: The specified output directory is not empty. This can be overridden with the --force flag: $params.outputDir.
  - WARNING: [ATAC] Overriding errors in the config file. Only TIs specified in the config file will be used.
  - WARNING: [ATAC] Undetermined FASTQ read files will be ignored.

Bio-Rad Laboratories, Inc.

Life Science Group

Website *bio-rad.com* ***USA*** *1 800 424 6723* ***Australia*** *61 2 9914 2800* ***Austria*** *00 800 00 24 67 23* ***Belgium*** *00 800 00 24 67 23* ***Brazil*** *4003 0399*
Canada *1 905 364 3435* ***China*** *86 21 6169 8500* ***Czech Republic*** *00 800 00 24 67 23* ***Denmark*** *00 800 00 24 67 23* ***Finland*** *00 800 00 24 67 23*
France *00 800 00 24 67 23* ***Germany*** *00 800 00 24 67 23* ***Hong Kong*** *852 2789 3300* ***Hungary*** *00 800 00 24 67 23* ***India*** *91 124 4029300* ***Israel*** *0 3 9636050*
Italy *00 800 00 24 67 23* ***Japan*** *81 3 6361 7000* ***Korea*** *82 2 3473 4460* ***Luxembourg*** *00 800 00 24 67 23* ***Mexico*** *52 555 488 7670*
The Netherlands *00 800 00 24 67 23* ***New Zealand*** *64 9 415 2280* ***Norway*** *00 800 00 24 67 23* ***Poland*** *00 800 00 24 67 23* ***Portugal*** *00 800 00 24 67 23*
Russian Federation *00 800 00 24 67 23* ***Singapore*** *65 6415 3188* ***South Africa*** *00 800 00 24 67 23* ***Spain*** *00 800 00 24 67 23* ***Sweden*** *00 800 00 24 67 23*
Switzerland *00 800 00 24 67 23* ***Taiwan*** *886 2 2578 7189* ***Thailand*** *66 2 651 8311* ***United Arab Emirates*** *36 1 459 6150* ***United Kingdom*** *00 800 00 24 67 23*