



# ScaleBio Seq Suite: RNA

## Data Analysis Handbook

For Research Use Only.

## Legal Notices

Document 1325194, Rev A, May 2025  
© 2025 Scale Biosciences, Inc.

5601 Oberlin Dr., Suite 110  
San Diego, CA 92121  
<https://scale.bio/>  
[support@scale.bio](mailto:support@scale.bio)

Scale Biosciences, Inc (“ScaleBio”). All rights reserved. No part of this document may be reproduced, distributed, or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without the prior written permission of ScaleBio. This document is provided for information purposes only and is subject to change or withdrawal by ScaleBio at any time.

### Disclaimer of Warranty:

TO THE EXTENT PERMITTED BY APPLICABLE LAW, SCALEBIO PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL SCALEBIO BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENT, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF SCALEBIO IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. Any warranties applicable to the ScaleBio products are set forth in the Terms and Conditions accompanying such product and such Terms and Conditions are not modified in any way by the terms of this notice.

### Trademark Information:

ScaleBio may make reference to products or services provided by other companies using their brand names or company names solely for the purpose of clarity, and does not assert any ownership rights over those third-party marks or names. Images were created with BioRender.com

### Patent Information:

ScaleBio products may be covered by one or more patents as indicated at: <https://scale.bio/legal-notice/>

### Terms and Conditions:

The use of the ScaleBio products described herein is subject to ScaleBio's Terms and Conditions that accompany the product, or such other terms as have been agreed to in writing between ScaleBio and the user.

### Intended Use:

All products and services described herein are intended FOR RESEARCH USE ONLY and NOT FOR USE IN DIAGNOSTIC PROCEDURES.

## **Table of Contents**

<b><i>Legal Notices.....</i></b>	<b><i>2</i></b>
<b><i>Introduction.....</i></b>	<b><i>4</i></b>
<b><i>Compatibility.....</i></b>	<b><i>5</i></b>
<b><i>Quick Start Guide.....</i></b>	<b><i>7</i></b>
<b><i>Chapter 1: Pipeline Setup, Installation and Testing.....</i></b>	<b><i>8</i></b>
1.1. Requirements.....	8
1.2. Install Nextflow .....	8
1.3. Install the ScaleBio Seq Suite: RNA Pipeline.....	8
1.4. Determine a dependency install method.....	8
<b><i>Chapter 2: Input Files.....</i></b>	<b><i>12</i></b>
2.1. Sequencing Reads.....	12
2.2. Reference Genome .....	13
2.3. Sample Barcode Table (samples.csv).....	14
<b><i>Chapter 3: Step-by-Step Overview of the Pipeline.....</i></b>	<b><i>16</i></b>
3.1. FASTQ Generation with ScaleRna.....	16
3.2. Read Trimming .....	17
3.3. FastQC.....	17
3.4. Barcode Parsing.....	17
3.5. Sample Demultiplexing.....	17
3.6. Generation of Library QC Report.....	17
3.7. Genome Alignment .....	18
3.8. Transcript Match .....	18
3.9. Unique Transcript Counts.....	18
3.10. Cell Filtering.....	18
3.11. Generation of Gene Expression Matrix.....	19
3.12. Generation of Sample QC Report.....	19
<b><i>Chapter 4: Overview of Analysis Output Files.....</i></b>	<b><i>20</i></b>
<b><i>Appendix A: Merging Multiple Runs.....</i></b>	<b><i>21</i></b>
<b><i>Appendix B: Software Dependencies.....</i></b>	<b><i>22</i></b>
<b><i>Appendix C: ScalePlex.....</i></b>	<b><i>23</i></b>
<b><i>Appendix D: Custom Reference Genome.....</i></b>	<b><i>29</i></b>
D.3 Creating a New Genome Index .....	30
D.4 Adding Extra Sequences or Transgenes .....	30
<b><i>Document Revision History.....</i></b>	<b><i>32</i></b>

## Introduction

The QuantumScale Single Cell RNA Kit uses a combinatorial indexing strategy to resolve gene expression data at single cell resolution by using a combination of unique RT Barcodes, Barcoded Beads, and PCR barcodes. The ScaleBio Seq Suite: RNA Data Analysis Pipeline is designed as an end-to-end workflow that takes users from raw sequencing output of their QuantumScale Single Cell RNA Kit library to a thorough assessment of that library's performance, cell calling, and ultimately generation of gene expression matrices.

This handbook serves as a high-level guide for setting up, running, and understanding the outputs of the ScaleBio Seq Suite: RNA Data Analysis Pipeline. For specific step-by-step instructions on installing and running the pipeline please refer to our [GitHub repository](#). The introductory readme markdown file ([README.md](#)) provides an overview of the workflow. Additionally, there is a series of markdown files (\*.md) within [ScaleRna/docs](#) to help guide users in more detail at each major step.

## Compatibility

Please confirm that you are using the correct version of the Seq Suite: RNA Data Analysis Pipeline compatible with the chemistry version of your Single Cell RNA Sequencing Kit (Table 1).

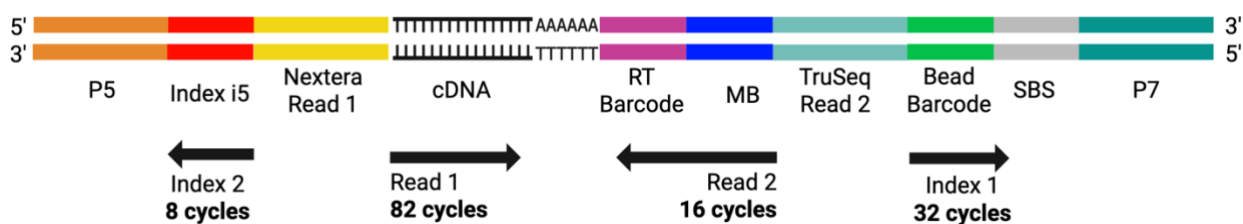
*Table 1: ScaleBio Pipeline and Assay Chemistry Compatibility*

Single Cell RNA Sequencing Kit Version	Seq Suite: RNA Data Analysis Pipeline Version	Note
Quantum v1	v2.0.0 or later	<ul style="list-style-type: none"> <li>QuantumScale Support</li> </ul>
Single Cell RNA Sequencing Kit v1.1	v1.6.2 or later	<ul style="list-style-type: none"> <li>ScalePlex Support</li> </ul>
Single Cell RNA Sequencing Kit v1.1	v1.4 or later	<ul style="list-style-type: none"> <li>i5 or i7 cell indexing</li> <li>Multi-plate merging support for Extended Throughput</li> </ul>
Single Cell RNA Sequencing Kit v1.0	v1.3.3 or later	<ul style="list-style-type: none"> <li>Only i7 cell indexing support</li> <li>No Extended Throughput support</li> </ul>

QuantumScale:

Quantum RNA libraries have an entirely different structure to that of the Single Cell RNA Sequencing Kit v1.1 or v1. With the new design alongside the massive increase in dataset size, Quantum library analysis will be restricted to Seq Suite: RNA Data Analysis Pipeline version 2.0.0 or later, and with the `--quantum` parameter set to true.

*Figure 1: Quantum RNA Library Design*

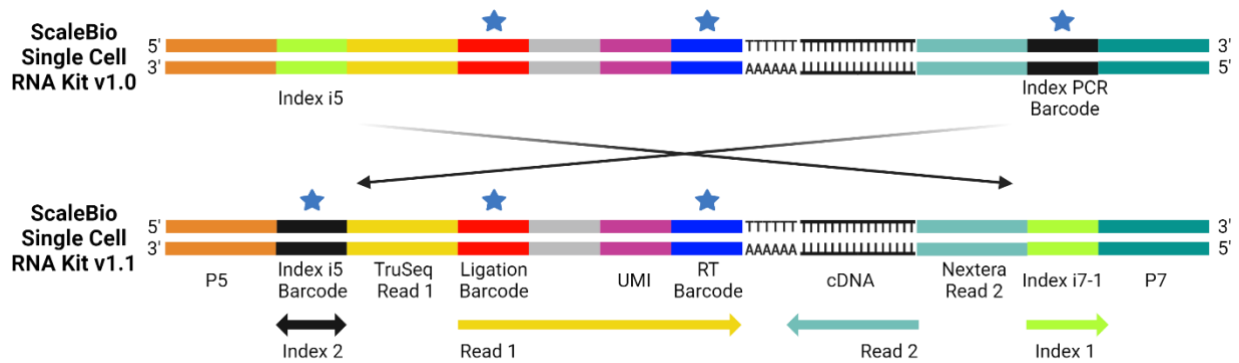


Single Cell RNA Sequencing Kit:

This compatibility requirement for v1.0 and v1.1 is due to different cell indexing configurations between the two kit versions. Cells are still indexed via unique RT Barcodes and Ligation Barcodes, but the position of the third barcode used for combinatorial indexing is different as illustrated Figure 2:

- **Single Cell RNA Sequencing Kit v1.0:** the third cell indexing barcode is Index PCR Barcode at the position of Index 1.
- **Single Cell RNA Sequencing Kit v1.1:** the third cell indexing barcode is Index i5 Barcode at the position of Index 2.

*Figure 22: Single Cell RNA Sequencing Kit RNA Library Design Comparison Highlighting the Barcodes Used for Combinatorial Indexing (Blue Star)*



In addition, changing the cell indexing configuration enables increasing the number of recovered cells from the assay by the usage of multiple Final Distribution Plates from the ScaleBio Single Cell RNA Extended Throughput v1.1 Kit. Version 1.4 and onwards of the Seq Suite: RNA Data Analysis Pipeline supports merging of multiple Final Distribution Plates.

Please review the [changelog](#) on GitHub on how to run multiple Final Distribution Plates through the pipeline.

## Quick Start Guide

### System Requirements:

- Linux system with GLIBC  $\geq$  2.17 (such as CentOS 7 or later)
- Java 11 or later
- 64GB of RAM and 12 CPU cores

### Instructions:

1. Install [Nextflow](#) (23.10 or later).
2. Download the latest [ScaleRna](#) workflow to your machine.
3. Select a *profile* to install [dependencies](#).
4. Launch the ScaleBio downsampled pipeline [test](#) run.
5. Download one of the ScaleBio pre-built [genomes](#) or generate a new [custom genome reference](#).
6. Create a sample barcode table ([samples.csv](#)) for ScaleBio sample demultiplexing.
7. Specify all [analysis parameters](#) in the [runParams.yml](#).
  - 7.1. *fastqDir* / *runFolder* and *genome*
  - 7.2.
    - 7.2.1. *runFolder* : *path/to/bcl/runFolder*
8. Setup your final [nextflow command](#) and launch the workflow!

## Chapter 1: Pipeline Setup, Installation and Testing

### 1.1. Requirements

The workflow can be launched either on a macOS or on a Linux system, such as CentOS 7 or later (GLIBC  $\geq 2.17$ ) with Java 11 (or later). For running the pipeline on either system, the recommended minimal configuration is 64 GB of RAM and 12 CPU cores. For smaller datasets, 32 GB of RAM and 6 CPU cores can be sufficient. In addition, the workflow requires temporary storage for intermediate files, ranging up to 3 TB for large (e.g. NovaSeq S4) sequencing runs.

### 1.2. Install Nextflow

To install the pipeline on a new system, first install Nextflow, following the instructions at <https://www.nextflow.io/>. The installation requires Java version 11 or higher. Once downloaded to your system, you can confirm that the Nextflow executable works as expected by running the following command:

```
nextflow run hello
```

Note that Nextflow can be installed in your user directory without admin rights on the system. Once you have the Nextflow command running, it is recommended to add the install location to the path variable in the users bash startup file (i.e., `.bash_profile`, `.bashrc`, `.profile`).

### 1.3. Install the ScaleBio Seq Suite: RNA Pipeline

The pipeline can be downloaded by two methods:

1. By going to the [ScaleBio Seq Suite: RNA GitHub page](#), clicking the green “Code” button and then “Download ZIP”. Unpack this file on your system directly in the directory in which you want to install the pipeline. To make sure the download is complete, make sure the Python files (.py extension) in the ScaleRna/bin directory have the appropriate execute privileges on your server.
2. The GitHub repository can be cloned to your machine:

```
git clone https://github.com/ScaleBio/ScaleRna.git
```

### 1.4. Determine a dependency install method

The workflow requires several dependencies (see [Appendix B: Software Dependencies](#)). With the ``-profile`` setting, users can select the method of dependency installment. For Nextflow there are multiple methods for installing dependencies. This allows users flexibility depending on their systems requirements or limitations. These dependency profiles can be selected in one of three options:

1. Docker (RECOMMENDED): Using containers (``docker``).
2. Singularity: Alternative container option, usually available as a module on most high-performance computing systems.
3. Conda: Using the conda package manager so Nextflow can generate conda environments.



If none of the above options are successful, there is a final option to manually [download](#) and install all necessary dependencies, as well as placing them on path. This option is prone to error and will usually require a lot more setup and installation time than the above options.

#### 1.4.1. Using Containers

Containers have all the necessary code, libraries, and configurations for the workflow to operate on most systems. If your system supports execution of a container (either using *Docker* or *Singularity*) this is likely the easiest way to handle dependencies, especially if you are familiar with running containerized workflows.

#### 1.4.2. Docker vs. Singularity Container

*Docker* support is enabled with the Nextflow command-line option `-profile docker`. Note that setting up *Docker* support for the first time on a new system typically requires admin (root) access and some familiarity with the system.

If your system access does not support the use of *Docker*, then *Singularity* is an alternative for container execution that is available on many HPC clusters. We require Singularity 2.3 or newer. Setting `-profile docker,singularity` (no space) will use the Singularity engine for all dependencies.

Container usage can require some extra setup to ensure all relevant file paths are accessible from inside the containers:

- For *Docker*, Nextflow will set the relevant options automatically at runtime to mount (bind) input and output paths to the container.
- For *Singularity*, this requires `USER BIND CONTROL` to be enabled in the system-wide configuration (see [The Singularity Config File](#) and the notes in the [Nextflow singularity documentation](#)).
- The environment variable `NXF_SINGULARITY_CACHEDIR` can be used to control where *Singularity* images are stored. This should be a writable location that is available on all compute nodes.
- Similarly, `TMPDIR` should be changed from the default `/tmp` to a location writable from the container if necessary.

#### 1.4.3. Using conda

Another option is using the [conda](#) package manager. Nextflow can automatically create conda environments with most of the needed dependencies. This mode is selected by setting `-profile conda`. In this case, the following additional steps need to be completed:

- Install ScaleBio Tools
  - These are ScaleBio specific programs that are currently not available through conda.
  - Run `/PATH/TO/ScaleRNA/envs/download-scale-tools.sh`

- This will install the pre-compiled binaries in `ScaleRNA/bin` (inside the Nextflow workflow directory), from where they will be available during workflow execution.
- If running from a sequencer runFolder (BCL) Illumina [BCL Convert](#) 3.9.3 is required to be installed (and available on `$PATH`).

See the [Nextflow documentation](#) for additional details of conda support in Nextflow.

#### 1.4.4. Manual Dependency Installation

As a final alternative, the required dependencies can be installed directly, either by hand or using conda. A list of all requirements can be found in Chapter 5 or in the environment `conda.yml` files in `ScaleRNA/envs`. Once you have installed the dependencies, and made sure they are available on `$PATH`, the workflow can be run without any `-profile`.

#### 1.4.5. Run the Workflow Test

A downsampled dataset including usage instructions can be found [here](#). Running this small sample dataset will allow you to rapidly test that you have installed the workflow correctly and your system requirements are met. This may also give you the opportunity to check compatibility of processed files with any downstream/secondary analysis.

NOTE: The raw sample data is stored on Amazon Web Services (AWS S3) and will be downloaded to your system automatically once you execute the Nextflow command (see [here](#)). Review your systems documentation for any additional steps that are required to download from AWS. Our test workflow reference genome and fastq files can be downloaded without an aws account or aws login credentials using the following commands [here](#).

#### 1.4.6. Workflow Parallelism

Individual steps of the pipeline are run as separate Nextflow tasks. Nextflow will launch as many parallel tasks concurrently as possible given available resources. The number of tasks can be controlled by the `-maxForks` nextflow parameter. Additionally, for many tasks, such as STAR alignment, the number of parallel threads inside a single task is configured dynamically to match available resources.

The `--splitFastq` option, on by default, enables an extra level of parallelism, based on the inputs provided. When starting from a bcl runFolder, the fastq files are split into individual PCR barcode-bead1 barcode files and the majority of processes downstream are subsequently processed by barcode. When starting from fastq files, the workflow will first confirm the PCR index used for library generation and execute barcode parsing in the same manner as starting with bcls with the resulting sample demultiplexed unaligned bam files being split by each individual PCR index by RT barcode. All resulting read groups are merged in the end to produce the same outputs per sample as in the default case.

NOTE: When using `--splitFastq` and `-fastqOut` the resulting fastq files will be saved in the PCR barcode split format and separated by samples. If desired, these files can be merged.

#### 1.4.7. Running the ScaleRna nextflow workflow as a job with an executor/scheduler

The dynamic allocation of resources in Nextflow can be selected with the “executor” configuration scope (see [Executors — Nextflow 23.04.1 documentation](#)). Nextflow also provides extensive documentation for HPC users such as SLURM or Moab. for hpc users such as SLURM or Moab.

[https://www.nextflow.io/blog/2021/5\\_tips\\_for\\_hpc\\_users.html](https://www.nextflow.io/blog/2021/5_tips_for_hpc_users.html)

<https://www.nextflow.io/blog/2021/5-more-tips-for-nextflow-user-on-hpc.html>

NOTE: By default (“*local*” executor), all tasks are launched on the same machine that Nextflow itself was started on, limiting the parallelism to the resources available on that computer (CPU cores and memory). Users with the minimum required resources should anticipate fold differences in runtime without use of executor function.

We recommend contacting your system's IT help desk to check if there is a nf-core nextflow [module](#) available for testing. These modules are rigorously tested and should provide guidance for the final nextflow.config setting and parameters. On AWS, Azure, or Google Cloud it can use *Batch* to achieve massive parallelism. The Nextflow [documentation](#) contains details about how each of these use cases can be configured.

## Chapter 2: Input Files

To run the pipeline the user will need to provide 3 inputs:

1. Sequencing reads
  - either bcl run folders, fastq files (R1, R2, and I1, I2), or unaligned CRAM files post Trimmer processed from Ultima Sequencing
2. Reference genome (STAR index)
3. Sample barcode table (samples.csv)

### 2.1. Sequencing Reads

**NOTE: Illumina i7 and i5 index reads are required to be exported in fastq format. See below on how to generate index reads.**

There are two formats reads can be used as input for the pipeline ([ScaleRna/Fastq Generation](#)):

#### 2.1.1. BCL files as input – *preferred pathway*

If the ScaleBio Quantum RNA library was sequenced alone in a sequencing run or on a flow cell lane, the easiest way to run the analysis is to start directly from the sequencer RunFolder [ `--runFolder`] (this is the sequencer output folder containing the *RunInfo.xml* file). In this case, the ScaleBio RNA workflow will internally generate FASTQ files appropriate for the pipeline input using Illumina *bcl-convert* ([BCL Convert Support \(illumina.com\)](#)).

#### 2.1.2. FASTQ files as input [Read 1, Read 2, Index 1 (i7), Index 2 (i5)]

If the raw sequencer output is not available as BCL files, or the ScaleBio RNA library was multiplexed with other libraries during sequencing, the analysis can be started from FASTQ files generated ahead of time. In this case, please note the following:

- INDEX READS: Generating index reads in fastq format can be done using the *CreateFastqForIndexReads* option in Illumina *bcl-convert* OR by using ScaleBio provided samplesheets (see below)
- Each ScaleBio Quantum RNA library has 384 unique i7/i5 index combinations per index PCR reaction used in the workflow for fastq generation. Each index PCR reaction corresponds to a base-balanced pool of 4 sequences. Importantly, index1 (i7) is the full bead barcode for that molecule. For demultiplexing, we only use a portion of that because it allows for us to not explicitly supply the full bead barcode space in fastq generation. Using ScaleBio samplesheets users should expect 96 X 4 fastq files (384 R1, R2, I1 and I2) per index pcr library, a number that will vary depending on the configuration of the kit used. Operating on a large number of files is optimal for ScaleRna and will result in decreased runtimes relative to less files but the same amount of total reads.
- Premade samplesheets (located here: [\[docs/examples/fastq-generation\]](#)) are included in the ScaleRna pipeline and can be used for FASTQ generation with *bcl-convert*. Using these will create the FASTQ files with names as shown below for a full ScaleBio run. The

resulting files will have the name of the Index PCR used as a prefix and alias of the portion of the bead barcode used.

```
QSR-1_1A_L001_R1_001.fastq.gz
QSR-1_1A_L001_R2_001.fastq.gz
QSR-1_1A_L001_I1_001.fastq.gz
QSR-1_1A_L001_I2_001.fastq.gz
```

- When generating fastq for multiple index PCR reactions, the same logic above applies with each index PCR yielding 96 X 4 fastq files (384 R1, R2, I1 and I2) per library sequenced. The resulting files will have the **Index PCR** prefix (**adapter primer set**) and **partial bead barcode alias**.

```
QSR-1_1A_L001_R1_001.fastq.gz
QSR-1_1A_L001_R2_001.fastq.gz
QSR-1_1A_L001_I1_001.fastq.gz
QSR-1_1A_L001_I2_001.fastq.gz
```

```
QSR-2_1A_L001_R1_001.fastq.gz
QSR-2_1A_L001_R2_001.fastq.gz
QSR-2_1A_L001_I1_001.fastq.gz
QSR-2_1A_L001_I2_001.fastq.gz
```

## 2.2. Reference Genome

To generate a STAR (v2.7 or higher) reference genome index, users require their respective genome fasta (.fa) file and gene annotation file (.gtf). Once the reference genome index is generated, these files are specified using a [genome.json file](#), which contains the file paths and other parameters. Further information can be found here: [Scale Reference Genomes](#). ScaleBio has pre-built genomes for human (hg38), mouse (mm39), and a mixed human/mouse barnyard genome are available here:

- <http://scale.pub.s3.amazonaws.com/genomes/rna/grch38.tgz>
- <http://scale.pub.s3.amazonaws.com/genomes/rna/mm39.tgz>
- [http://scale.pub.s3.amazonaws.com/genomes/rna/grch38\\_mm39.tgz](http://scale.pub.s3.amazonaws.com/genomes/rna/grch38_mm39.tgz)

Download the appropriate reference file to your system, unpack (`tar -xzf GENOME.tgz`), and then specify the JSON file inside the directory (e.g. `grch38/grch38.json`) for the analysis [with the command line: `--genome`; or in the `runParams.yml`].

**NOTE:** You must download and unpack these files locally first. Do not specify the URLs to these TGZ files directly in the `--genome` option. Similarly, DO NOT use the example `genome.json` ([docs/examples/genome.json](#)) for real analysis beyond the test run. This test run aligns to a truncated version of the genome that is in place to ensure the pipeline is properly working in the user's environment and should not be used for analysis.

Building a new/custom STAR reference index for a different species or adding additional sequences/annotations requires the genome sequence (FASTA) and gene annotation (GTF). Adding additional sequences or modifying our pre-built genomes still requires a new reference genome build.

**NOTE:** When adding additional annotations to our pre-built genomes, please abide by the pre-built genome .gtf formatting including case-matching to ensure proper report generation.

Please see below the commands for generating a STAR index (v2.7 or higher) and new JSON file:

1. Build a STAR index, using at least STAR version 2.7:

```
STAR --runMode genomeGenerate --runThreadN 16 --genomeDir star.ref --genomeFastaFiles  
GENOME_SEQUENCE.fa --sjdbGTFfile GENE_ANNOTATION.gtf
```

2. Generate a genome.json file with the following as a template:

```
{  
  "name": "GRCh38",  
  "speciesName": "Homo sapiens",  
  "star_index": "path/to/star.ref",  
  "gtf": "path/to/GENE_ANNOTATION.gtf",  
}
```

NOTE: When creating a custom genome, all features included in the GTF file will be used for gene expression analysis by the workflow. To exclude certain annotations (e.g. pseudo-genes), filter the GTF file before generating the STAR index.

### 2.3. Sample Barcode Table (samples.csv)

The ScaleRna Quantum library represents a pool of cells and needs to be further separated into sample specific files for analysis. **A sample here is the groupings of cells defined by their organization on the RT plate well position.** A sample barcode table (e.g. [EXAMPLES](#)) file is used to list the samples included in an analysis run, their sample barcode (RT) sequences and optional sample-specific analysis parameters.

It is a comma separated file (CSV) with a header line (column names), followed by one sample per line. Please see below the minimum requirements:

*Table 2: Contents of the sample barcode table entries*

sample	barcodes	libIndex2
Enter your desired sample name here	Enter the ScaleBio RT barcodes for the sample row	Names of RNA library index pools used for library generation and sequencing

**Notes on the “barcodes” column used for demultiplexing samples**

During analysis, sequencing data is first converted into library FASTQ files (controlled by optional libIndex2 column). If multiple samples were included in one sequencing library, the pipeline will further demultiplex based on the sample (RT) barcodes. These sample barcodes are defined by the wells on the RT Barcode Plate, for example see the table below.

*Table 3: Example sample barcode table with 2 samples*

sample	barcodes	libIndex2
Jurkat-DMSO	1A-6H	QSR-1;QSR-2;QSR-3;QSR-4;QSR-5;QSR-6'QSR-7;QSR-8
Jurkat-500nM	7A-12H	QSR-1;QSR-2;QSR-3;QSR-4;QSR-5;QSR-6'QSR-7;QSR-8

**libIndex2 specification (optional)**

- Users have the option to supply which specific library index PCR primer was used in their Quantum analysis in the libIndex2 column of the sample barcode table.
- The full list of sequences and their names can be found here: [Quantum PCR Pools](#). The pipeline accepts one or multiple names supplied (like in the example above). If using multiple, the names should be delimited by a ; character
- If the user does not supply this explicitly, depending on input mode, the pipeline will do one of the following:
  - BCL files as input
    - The pipeline will attempt to demultiplex into every potential library index set, and will then filter out any that have little to no reads associated with them for analysis
  - Fastq files as input
    - The fastq files will be checked in the workflow for which PCR library index is detected in each file \*this behavior assumes that each index2 file only has one specific pool of index PCR sequences in it, found in [Quantum PCR](#)

[Pools](#) and will carry through those files that have valid matches into downstream analysis

Rules to abide by for naming samples, RT pools and libraries:

- “sample” should consist ONLY of letters, numbers, dash (-) and dot (.).
- Underscores are not valid.
- “barcodes” is required if more than one sample is processed on the RT Barcode Plate.

Rules to abide by for listing RT or ScalePlex well barcodes:

The RT wells used for each sample are given as either:

- An individual value: 1A
- A range of wells: 1A-2H
- A list of values or ranges, separated by semicolon (;): 1A;2A-2D

All ranges are read in column-wise order,

- e.g. 1A-2C, refers to 1A-1H (all of column 1) plus 2A, 2B and 2C.

Samples plated by rows they should input each RT well separated by a semi-colon

- e.g. Row 1 = 1A;2A; 3A;4A;5A;6A;7A;8A;9A;10A;11A;12A
- e.g. Row 4 = 1D;2D; 3D;4D;5D;6D;7D;8D;9D;10D;11D;12D

## Chapter 3: Step-by-Step Overview of the Pipeline

In this chapter, we list the steps performed by the sequencing analysis pipeline, showing the order in which they are performed, and providing high-level information about the analyses performed at each step.

### 3.1. FASTQ Generation with ScaleRna

- This is performed only if a RunFolder (BCL files) is provided as input, rather than FASTQ files that were pre-generated using Illumina *bcl-convert*.
- All necessary FASTQ files are generated, including the i7 and i5 index reads (I1 and I2, respectively)
  - One set of FASTQ files, containing the raw unaligned reads, is generated for the entire sequencing run/lane.
  - Reads not matching one of the expected ScaleBio RNA i5 PCR barcode sequences are filtered (into the *Undetermined* FASTQ files).
  - Sequencing ReadsIf --splitFastq is enabled, a separate FASTQ file is produced for each PCR index pool X partial bead barcode (i7) resulting in 96 sets of files, that can be processed in parallel and merged at the end of the workflow.
- The *bcl-convert* step produces the standard Illumina reports on the number of reads per library and related metrics: [Output Files \(illumina.com\)](https://illumina.com/output-files)



### 3.2. Read Trimming

- Trims any transcript reads that run into a Poly-A, Poly-G stretch or additional adapters using *cutadapt* ([Cutadapt – Cutadapt 4.2 documentation](#)).
- Cuts any read at the first occurrence of a stretch of 7 As within 8 bp.

### 3.3. FastQC

- This is an optional step to run QC reports on the input FASTQ files [`--fastqc`] using *FastQC* ([Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data.](#)).
- The FastQC reports include information on output such as Q30 scores, adapter content, and base distribution for Read 1 & 2.
- The index reads are excluded from the report.

### 3.4. Barcode Parsing

- In this step, the single-cell barcodes are extracted from the reads and error corrected using ScaleBio *bcParser* executable.
- This is included in the Docker/Singularity container or can be downloaded with [ScaleRNA/env/download-scale-tools.sh](#)
- Error correction is done against the list of expected barcode sequences, allowing up to 1 mismatch.
- If `--splitFastq` is enabled, each set of FASTQ files (e.g. each sequencing lane) is processed in parallel, otherwise they are pooled before this step.

### 3.5. Sample Demultiplexing

- If different samples were loaded into different wells of the RT Barcode plate during the ScaleBio Single Cell RNA Sequencing workflow, they are separated at this point for independent analysis.
- Implemented in *bcParser* together with Barcode Parsing.
- Uses the *samples.csv* sheet (see [Chapter 2: Input Files](#) for more information) to define RT barcodes for each sample.
- Outputs an unaligned bam file (cell barcodes in the tags, transcript in read) for each sample.
  - Transcript reads shorter than 16 bp after trimming are discarded.
- If `--splitFastq` is enabled and the reads were not already split on PCR barcode during *bcl-convert*, reads within each input FASTQ will be split based on RT barcode here. These subsets can then also be processed in parallel. RT barcodes corresponding to the same sample will be merged at the end of the workflow.

### 3.6. Generation of Library QC Report

- Produces an HTML report with QC metrics for the whole ScaleBio RNA library. This report focuses on barcode matching rates, read distribution across samples, and data quality across all barcodes (RT, Bead, and Index PCR).

### 3.7. Genome Alignment

- Aligns transcript reads to the genome using *STAR* ([STAR/STARsolo.md at master · alexdobin/STAR \(github.com\)](#)).
- STAR performs local sequence alignment, including spliced alignments to both known and novel transcripts / isoforms.
- Optionally outputs a sorted BAM file with alignments for custom downstream analysis or visualization [`--bamOut`].

### 3.8. Transcript Match

- Matches aligned reads to genes and transcripts for expression quantification.
- Included in *STARsolo* ([STAR/STARsolo.md at master · alexdobin/STAR \(github.com\)](#)).
- Matches reads to exons and introns, in sense direction only (relative to transcript annotation).
- Exon matches are prioritized over intron matches (50% read to exon overlap).
- Reads that match an exon in antisense direction are filtered, i.e., not counted even if they overlap an intron in sense direction.
- The gene annotation used is the one built into the STAR index (from the GTF used during index generation).

### 3.9. Unique Transcript Counts

- The number of unique transcripts for each cell and gene combination is counted.
- Included in *STARsolo* ([STAR/STARsolo.md at master · alexdobin/STAR \(github.com\)](#)).
- Transcripts are unique if they differ in Cell Barcode, mapped gene, or UMI sequence.
- If a read matches multiple genes, the count is divided among them in proportion to the unique counts of the genes. See [STAR/STARsolo.md at master · alexdobin/STAR · GitHub](#) for details ("PropUnique" method). This behavior can be controlled with the `--starMulti` option.
- By default, we allow for a read to map to up to 6 positions

### 3.10. Cell Filtering

- CellFinder (default)
  - CellFinder is an [EmptyDrops](#)-like cell calling approach. This option is enabled when the parameter *cellFinder* is true. Cells that are above the *minUTC* but below the provided or calculated *UTC threshold* can potentially be "rescued" based on expression differences from the ambient RNA.
- Unique Transcript Count Thresholding
  - Calculates per cell metrics and a unique transcript threshold to filter cell barcodes belonging to real cells from background.
  - The cell threshold is determined following the method in *STARsolo*:
    - A preliminary list of possible cells is set (either based on the number of expected cells set in *samples.csv*, or the number of cell-barcodes with over 100 [`--minReads`] unique transcripts.
    - TopCount is set to the unique transcript count of the top 1 percentile [`--topCellPercent`] of preliminary cells.
    - The threshold is set to TopCount / 10 [`--minCellRatio`].
  - Setting a minimum UTC threshold

- A unique transcript count threshold is used when the parameter *UTC* is greater than 0 and *cellFinder* is not enabled. In this case every barcode with a total count  $\geq UTC$  is called as a cell.

### 3.11. Generation of Gene Expression Matrix

- This output file is a sparse matrix (MTX file format) of the gene expression per cell (unique transcript counts).
- Both an unfiltered (all cell-barcodes) and a filtered (cell barcodes above the cell threshold) matrix are produced.
- Can be read into [Seurat](#), [ScanPy](#) and similar.
- For more information on the specific metrics found in this report please see the links in [Chapter 4: Overview of Analysis Output Files](#).

### 3.12. Generation of Sample QC Report

- A summary report with metrics for each sample.
- Includes mapping metrics, cell-counts, and sensitivity metrics.
- Includes distribution of RT barcodes for the specific sample.
- Produces a HTML document and a CSV file with metrics in text format.
- Note: for more information on the specific metrics found in this report please see the links in [Chapter 4: Overview of Analysis Output Files](#).

## Chapter 4: Overview of Analysis Output Files

For more information about the specific metrics found in any of the output files please see the links below which contain the most up-to-date information.

- [Outputs Overview](#)
- [QC Report Overview](#)

## Appendix A: Merging Multiple Runs

If the Index PCR Libraries were sequenced on separate sequencing runs, it is suggested to analyze each sequencing run's data individually, and then later merge in a separate Nextflow run.

- Create a separate samples.csv file for each sequencing run's data and process normally.**Error! Reference source not found.**
- Create another samples.csv that lists all samples and Index PCR's to be merged.
  - Add a new resultDir column that points to the workflow output directories for the individual runs.
- Run the workflow with the `--reporting` and `--merge` option.
  - `--reporting` uses the previous alignment results instead of re-processing all reads.
  - `--merge` creates merged outputs for each sample across all plates as before.
  - Do not specify any input reads at this step (no `--runFolder` or `--fastqDir`).

## Appendix B: Software Dependencies

- Java (11 or later)
- Nextflow (23.10 or later)
- Fastq generation:
  - bcl-convert 3.9.3
- For the full complement of dependencies, please visit our GitHub page: [ScaleBio Seq Suite: RNA](#)

## Appendix C: ScalePlex

### C.1. Overview

ScaleBio Seq Suite: RNA v1.6 (or later) provides support for the ScalePlex product within the ScaleRNA workflow. This is inclusive for extended throughput, and as of v2.0 or later Quantum as well. ScalePlex enables a higher degree of sample multiplexing by labeling material in fixation prior to loading onto the RT plate. Each well of the ScalePlex Oligo Plate is loaded with two ScalePlex oligos that, in combination, uniquely mark cells from that well. Each column and row have their own specific ScalePlex oligo associated, and their intersect per well dictates the expected combination which are used for counting and final assignment (represented in Figure C.1.). Cells from ScalePlex fixation are pooled, and this pool is then loaded into the RT plate.

- Processing of the pipeline still operates as treating the “sample” as the ScalePlex pool in this case. The sample barcode table (described below) still assumes the organization of the RT plate as the “samples” under this scheme.

These oligo labels then propagate through the indexing strategy alongside the mRNA material per cell and are amplified after the workflow is completed into their own uniquely indexed library. The i5 sequence of the resultant library is specific to the ScalePlex fraction while the i7 index sequence is the same as the mRNA and corresponds to the same bead barcode as the mRNA for the cell. (Figure C.2.)



Figure C.1. ScalePlex Oligo Plate diagram, illustrating combinatorial ScalePlex oligo scheme

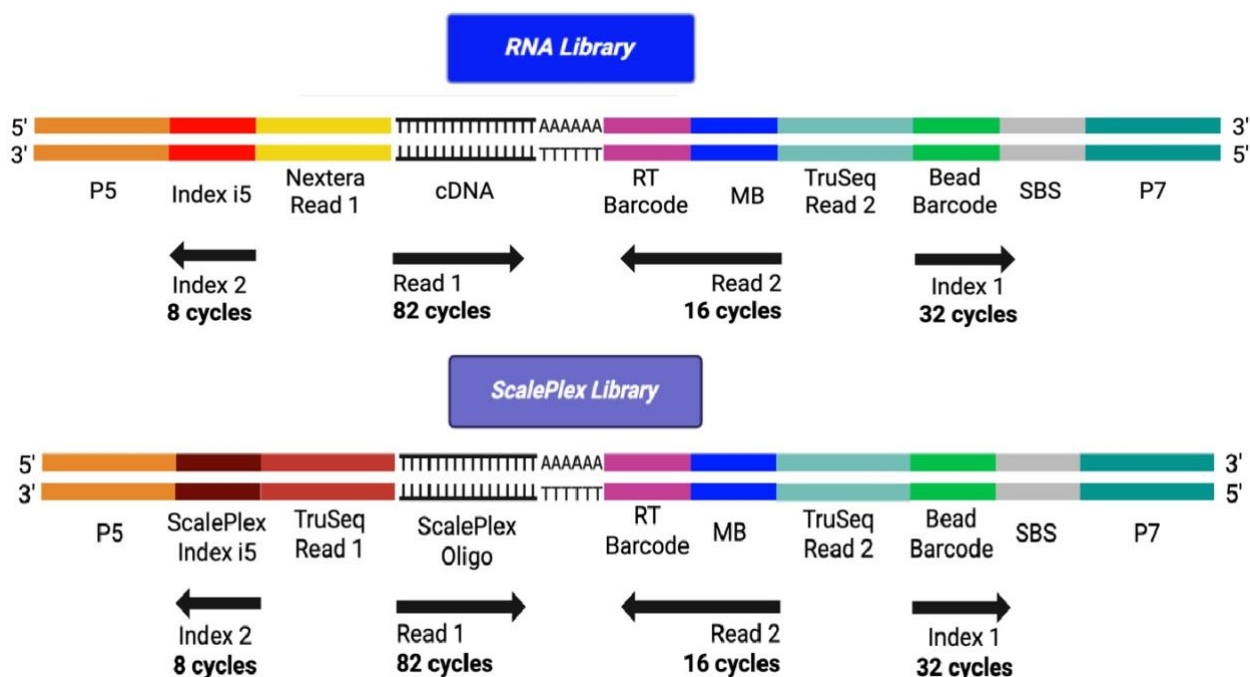


Figure C.2. RNA and ScalePlex Library Molecule Diagram

When processing the data, a handling of inputs splits sequencing reads into mRNA and ScalePlex library partitions that are then processed in parallel. The mRNA data is analyzed in the same way as our standard ScaleRNA workflow, but now with the processing of the ScalePlex library in tandem. The final outputs and reports will reflect the presence of the ScalePlex assignment for samples as defined by their RT well position. The ScalePlex library reads undergo the same barcode sequence validation as do the mRNA, but with an added layer to ensure that the ScalePlex target sequences are also present in the reads. This is then used to generate a UMI counts matrix per individual oligo per cell, and finally we use these counts for ultimate assignment per cell back to their ScalePlex Oligo Plate [position of origin](#).

## C.2. Inputs

Firstly, a workflow level parameter of “scalePlex” must be set to true either in the in-line execution or runParams.yml to ensure the workflow proceeds with proper analysis. The sample barcode table will also accept additional columns: scalePlexLibIndex2, and scalePlexBarcodes.

The same input requirements as discussed in [Chapter 2](#) pertaining to the RNA portion of the analysis are also maintained in this configuration, but now with the additional sequencing read data that correspond to your ScalePlex library paired with your RNA library. We support two main input modes for your sequencing data, BCL files direct from the sequencing run folder or fastq files discussed below.



### C.2.1. Changes to the Sample Barcode Table

The inclusion of the ScalePlex module for analysis in our ScaleRNA workflow has added several optional columns to the sample barcode table that can be engaged with to specify certain features regarding the ScalePlex Oligo Plate usage per sample as defined by RT, or for information regarding the sequencing reads used as input for the analysis.

**IMPORTANT NOTE:** In the non-scaleplex configuration, the “sample” defined in the sample barcode table refers to the groupings of cells from samples as organized on the RT plate. A user loads material from donor 1 into wells 1A-3H for example, and so their Donor1 sample has a barcodes value of 1A-3H for that row of the document. In the case of ScalePlex, the definition of a “sample” is still defined by the cell’s organization on the RT plate, but here what is being loaded is a POOL of samples mixed together. The pipeline will operate on this POOL of samples as one “sample”, then will perform scaleplex assignment back to the For the column pertaining to the sequencing read information, we’ll discuss in sections C.2.2. and C.2.3.

(scalePlexLibIndex2) of this handbook appendix. One additional optional column is independent of sequencing input mode, called “scalePlexBarcodes”. This allows for the user to specify which wells or columns of the ScalePlex Oligo Plate were used when generating the pool of samples loaded into each sample as defined by their RT plate well position. The convention for specifying this is exactly as how we specify the RT plate positions in the “barcodes” column, but critically here we are referring to the ScalePlex Oligo plate position rather than the RT plate position (Figure D.1.). See section [2.3](#) for further details regarding how that is encoded.

### C.2.2. BCL file input

This configuration assumes that your ScaleRNA and ScalePlex libraries were sequenced together in the same sequencing run. Here, if the “scalePlex” parameter is enabled, and the samples barcode table (samples.csv) is configured to reflect the RNA library’s RT sample designation, then with no additional modification the workflow will attempt to process the RNA and ScalePlex libraries. It does this by assuming that any of the potential ScalePlex library indices are present and will demultiplex reads for both RNA and ScalePlex and then subsequently split into samples as defined by RT for processing. Finally, the user may also supply a “scalePlexLibIndex” column to the file and explicitly supply the index sequence used in their study (references can be found here). There, the workflow will demultiplex based on the supplied sequence and proceed with downstream analysis.

sample	barcodes	libIndex2	scalePlexLibIndex2
PBMC-1	1A-6H	QSR-1;QSR-2	QSR-1-SCALEPLEX;QSR-2-SCALEPLEX
PBMC-2	7A-12H	QSR-1;QSR-2	QSR-1-SCALEPLEX;QSR-2-SCALEPLEX

The full list of i5 index sequences can be found at the following: [Quantum i5 scaleplex](#)

### C.2.3. Fastq file input

This configuration supports using already generated fastq files as input, one set of fastq's for your ScaleRNA Quantum library, and a set of fastq's for your ScalePlex library associated with the RNA experiment. Once the files are organized, in the sample barcode table (samples.csv) document used for input, the user can specify the libIndex2 used for RNA and ScalePlex respectively exactly as they would in the case of BCL file input, but now just supplying the path to the parent directory of the fastq files with the `-fastqDir` parameter

## C.3: Amended Trimming SOP for RNA in ScalePlex

The RNA portion of the workflow described in Chapter 3 remains untouched with the inclusion of ScalePlex with one important distinction. In the absence of ScalePlex, it is our default to first trim sequencing reads using cutadapt to remove bases corresponding to PolyA or PolyG stretches in the read. With the inclusion of the “scalePlex” set to True, we will additionally trim bases that correspond to the PCR handle of the scalePlex oligo in the RNA library. The goal of this trimming is to deplete any scalePlex molecules that could exist in the RNA fraction that could contribute to artificial alignment and quantitation. The design and modifications of the scalePlex oligo limit this fraction substantially, but for the purposes of data sanitation the trimming is still included.

## C.4. ScalePlex Assignment

### C.4.1. Assignment Methodology

As previously mentioned in C.1. Overview, the inclusion of ScalePlex to your library generation workflow involves the addition of two ScalePlex oligos, one specific to each row of the ScalePlex Oligo Plate and another specific to each column. In the resultant sample's analysis, we employ the oligo's unique combination per cell and use this to map back to the original well of the ScalePlex Oligo Plate that the cell was subjected to the labeling.

Once the ScalePlex oligo unique transcript counts are generated for each sample, we constrain the barcodes analyzed to be only those that passed cell calling in the RNA workflow and perform assignment. There are two methods of assignment that are supported by the workflow, controlled by the “scalePlexAssignmentMethod” parameter.

The default approach, when the parameter is set to “bg”, we build an expected background count estimation for each ScalePlex unique transcript count across the whole sample's cells (sample as defined by RT at this stage), and perform a statistical test of the observed counts against that background. If the top two detected ScalePlex labels both pass the background

and correspond to an expected combination of labels that align back to a well of the ScalePlex Oligo Plate, then the cell is assigned. Optionally, there is a “scalePlexPercentFromTopTwo” parameter that controls a threshold based on the ratio of ScalePlex unique transcript counts for any cell coming from the top two detected (given that we are doing a combinatorial based assignment, the top two should both be very enriched over total). By default this is set to zero, but provides the user a means to improve their assignment confidence to remove any higher background ScalePlex barcodes if need be.

The second method of assignment is a simpler approach that leverages the fold change of a cell’s second highest ScalePlex count compared to the third highest.

When “scalePlexAssignmentMethod” is set to “fc”, the workflow check to see if the fold change of the second highest ScalePlex count to the third highest ScalePlex count is > scalePlexFCThreshold parameter value (default 2) and if so the assignment proceeds. The same validation of expected pairs is also enforced as in the background based method.

**\*NOTE\*** We recommend using the default background based assignment approach for most cases. The fold change based assignment is best suited with datasets with low overall ScalePlex counts per cell, where the background estimation can run into issues in the testing.

#### C.4.2. Failure Modes for Assignment

While the assignment logic was described above in section D.4.1., there are several failure modes for cells to fail assignment and thus are unable to be associated with the ScalePlex Oligo Plate well of origin and are detailed below:

1. Max Fail
  - a. A cell fails assignment and is labeled “Max Fail” in the “bg” method when at least one of the top two ScalePlex oligos detected in that cell were not among those that passed the statistical background test for that cell
2. Indeterminate
  - a. A cell fails assignment and is labeled “Indeterminate” in the “bg” method when there were no ScalePlex oligo counts per cell that passed when compared versus the estimated background
3. Enrich Fail
  - a. A cell fails assignment and is labeled “Enrich Fail” in the “bg” method in the case that “scalePlexPercentFromTopTwo” is enabled and fails to pass the enrichment threshold set by the user supplied value.
  - b. A cell fails assignment and is labeled “Enrich Fail” in the “fc” method in the case that the fold change of the second highest ScalePlex oligo counts over the third highest ScalePlex oligo count was < 2.
4. Unexpected
  - a. A cell fails assignment and is labeled “Unexpected” when it passed all of the other assignment criteria for either the “bg” or “fc” method but the two passing ScalePlex oligos in combination do not match with a combination that either is:
    - i. not on the plate in the first place such as the top two being from two different columns rather than a combination of a row and column

- ii. If the user supplied an entry for “scalePlexBarcodes” for the sample in question and the assignment falls outside the constraint specified

The result of the assignment per cell can be found in the “assigned\_scaleplex” column of the SAMPLE.allCells.csv in the samples directory, more details can be found in the next section C.5. Outputs

## C.5. Outputs

The full list of outputs and descriptors can be found here: [LINK](#)

Briefly, engagement of the ScalePlex module for analysis will add an additional library qc report for the ScalePlex sequencing reads. Additionally, for each sample as defined by RT well designation (specified in the samples.csv) the sample specific report will feature a new “ScalePlex” tab that reports that summarize the performance of your ScalePlex sample material.

The main output that is relevant to downstream analysis is in the SAMPLE.allCells.csv, which contains per-cell metadata for the RNA fraction for each cell barcode in the analysis. A full description can be found in the link above, but it is worth mentioning specifically here that the “assigned\_scaleplex” column is the place in which cells have their ultimate assignment indicated when ScalePlex is enabled. When performing downstream analysis of ScalePlex data, the user should incorporate the SAMPLE.allCells.csv into the metadata of their analysis object and can then leverage the labels in the “assigned\_scaleplex” column to subset or summarize their analysis based on the true sample of origin based on the assignment.

## Appendix D: Custom Reference Genome

### D.1 Reference Genome

The workflow uses a genome reference and gene annotation for analysis. Reads are aligned to the full genome sequence and then matched to annotated transcripts using the RNA-seq aligner STAR. Hence specifically a STAR (v2.7.4 or higher) genome index is required, including both the sequence (fasta file) and gene annotation (GTF file).

### D.2 Pre-built Genomes

ScaleBio pre-built reference genomes for human (hg38), mouse (mm39), and a mixed human/mouse barnyard genome are available here:

- <http://scale.pub.s3.amazonaws.com/genomes/rna/grch38.tgz>
- <http://scale.pub.s3.amazonaws.com/genomes/rna/mm39.tgz>
- [http://scale.pub.s3.amazonaws.com/genomes/rna/grch38\\_mm39.tgz](http://scale.pub.s3.amazonaws.com/genomes/rna/grch38_mm39.tgz)

All of these use ENSEMBL annotations, subset to only “protein\_coding” and “lncRNA” and “IG\_?\_gene” biotypes.

Download the appropriate reference file to your system, unpack (`tar -xzf GENOME.tgz`), and then specify the JSON file inside the directory (e.g. grch38/grch38.json) for the analysis [with the command line: `--genome`; or in the runParams.yml].

**NOTE:** You must download and unpack these files locally first. Do not specify the URLs to these TGZ files directly in the `--genome` option. Similarly, DO NOT use the example genome.json ([docs/examples/genome.json](#)) for real analysis beyond the test run. This is a truncated version of the genome used only to ensure the pipeline is properly setup in the user’s environment and should not be used for real data.

## D.3 Creating a New Genome Index

An example command to build a STAR index:

```
STAR --runMode genomeGenerate --runThreadN 16 --genomeDir star.ref --
genomeFastaFiles Homo_sapiens.GRCh38.dna.primary_assembly.fa --sjdbGTFfile
Homo_sapiens.GRCh38.103.biotypeFiltered.gtf
```

- **star.ref** is the output directory which will contain the STAR index
  - The whole directory with all files is required during analysis
- **Homo\_sapiens.GRCh38.dna.primary\_assembly.fa** is the genome sequence in fasta format
  - Generally a reference genome with only ‘primary’ contigs (chromosomes) should be used, i.e. no alternative haplotypes or sequence patches (e.g. Ensemble ‘primary’)
- **Homo\_sapiens.GRCh38.103.biotypeFiltered.gtf** is the gene annotation in GTF format.
  - This has to match the genome sequence (fasta), i.e. same genome version and chromosome names (watch for chr1 vs. 1)
  - All genes included in the GTF file will be used for gene expression analysis. To exclude certain annotations (e.g. pseudo-genes), filter the GTF file before generating the STAR index.
  - STAR specifically uses “exon” annotations in the file, linked into transcripts by “transcript\_id” and grouped into genes by “gene\_id”. Optionally “gene\_name” is used to output human-readable gene-names. All fields are case sensitive!
- See “Generating genome indexes” in the STAR manual ([STAR/doc/STARmanual.pdf at master · alexdobin/STAR \(github.com\)](https://star-bioinformatics.github.io/doc/STARmanual.pdf)) for additional details and advanced options.

The ScaleRNA analysis workflow uses a [genome.json file](#), to specify the genome index location and other related parameters. To use a new reference genome, create a new json file with the path to the STAR index and GTF file and specify for the workflow with the `--genome`. See [Scale Reference Genomes](#) for additional details on the `genome.json` file format.

## D.4 Adding Extra Sequences or Transgenes

If the sample being analyzed includes a trans-gene or similar additional transcript being captured, a new reference genome including this extra sequence is required. This is true whether the extra transcript is targeted through the Poly-A tail or with a custom RT primer. In general this ‘alignment-based’ approach will work with one or multiple extra transcripts, as long as they are sufficiently unique from each other and the rest of the genome to be uniquely mappable with the RNA reads. Standard alignment is not optimal for short barcodes or similar sequences such as CRISPR guides; See [ScaleBio/ScaleCRISPR: ScaleBio Seq Suite: CRISPR Workflow \(github.com\)](#) for these use-cases.

First the sequence for the trans-gene should be added as an extra contig to the end of the fasta file, e.g.

```
>chr1
TAACCCTAACCCTAACC...
>chr2
...
>transgene1
ATGGTGAGCAAGGGCGAGGAGGATAACAT
```

Second an entry in the GTF file is needed. Typically this can simply be a single ‘exon’ spanning the entire extra sequence

```
transgene1      custom exon 1      1000      .      +      .      gene_id
"transgene1"; transcript_id "transgene1";
```

This will count all reads mapping to the transgene sequence as expression.

These fasta and GTF files can then be used to build a new STAR index and genome.json (see above). The extra transcript will be included in all outputs together with regular annotated genes.

## Document Revision History

Revision	Revision Date	Document ID	Changes
Rev A	May 2025	1325194	Initial release