Takara Bio USA

Cogent™ NGS Analysis Pipeline User Manual

Cat. Nos. Many
software v3.1
(011525)

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

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

I. Table of Contents

Table of Figures

Table of Tables

II. Introduction

Cogent NGS Analysis Pipeline (CogentAP) is a bioinformatics software for analyzing RNA-seq and DNA-seq stored in FASTQ files generated from libraries prepared using select Takara Bio next-generation sequencing (NGS) reagent kits. The output from CogentAP can then be imported into Cogent NGS Discovery Software (CogentDS) for additional processing and visualizing the data.

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

The program takes software-modified input files from sequencers and outputs the following:

- an HTML report, with results typical to single-cell analysis,
- an R data object (rds file) which can be used as input for CogentDS, and
- processed data files, such as a gene matrix and stats files, which can be used for further analysis (for kits other than the Shasta™ Total RNA-Seq Kit and for analyses with ≤5,000 barcodes).

CogentAP uses Nextflow as the pipeline manager and can be run on a Linux server using a command-line interface.

A. What's New

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

- **Cogent NGS Analysis Pipeline v3.1**
  - Changes to the install and upgrade procedure
  - New pipeline framework (Nextflow)
  - New DNA-seq analysis workflow, including CNV calling
  - New DNA-seq demux option
  - Option to perform ribodepletion of data
  - New experiment type options for the following kits:
    - Shasta Total RNA-Seq Kit - 2 Chip
    - Shasta Whole-Genome Amplification Kit - 2 Chip

NOTE: Release notes for prior versions can be found on the Cogent NGS Analysis Pipeline product page.

B. Supported NGS Products

Table 1 lists the Takara Bio products that the sequencing results can be processed by CogentAP. For processing sequencing results from Takara Bio immune profiling kits, refer to the Cogent NGS Immune Profiler.

Table 1. Applications and kits compatible with Cogent NGS Analysis Pipeline v3.1.

System	Experiment type	Kit or application
Shasta Single Cell System	Single-cell full gene-body total RNA-seq analysis	Shasta Total RNA-Seq Kit - 2 Chip
	Single-cell whole-genome amplification	Shasta Whole-Genome Amplification Kit - 2 Chip

System	Experiment type	Kit or application
ICELL8® cx Single-Cell System	Single-cell full gene-body total RNA-seq analysis	Shasta Total RNA-Seq Kit - 2 Chip
	Single-cell whole-genome amplification	Shasta Whole-Genome Amplification Kit - 2 Chip
	Single-cell full gene-body transcriptome analysis	SMART-Seq® Pro Application Kit - 2 Chip
Plate-based	Single-cell full gene-body transcriptome analysis (with UMIs)	SMART-Seq mRNA LP (with UMIs)
	Single-cell full gene-body transcriptome analysis (no UMIs)	SMART-Seq mRNA LP SMART-Seq mRNA SMART-Seq mRNA Single Cell LP SMART-Seq mRNA Single Cell SMART-Seq mRNA HT LP SMART-Seq mRNA HT SMART-Seq v4 PLUS Kit* SMART-Seq v4 Ultra® Low Input RNA Kit* SMART-Seq Single Cell PLUS Kit* SMART-Seq Single Cell Kit* SMART-Seq HT PLUS Kit* SMART-Seq HT*
	Strand-specific total RNA-seq for mammalian samples (with UMIs)	SMART-Seq Total RNA Pico Input with UMIs (ZapR® Mammalian) SMARTer® Stranded Total RNA-Seq Kit v3 - Pico Input Mammalian*
	Strand-specific total RNA-seq for mammalian samples (no UMIs)	SMART-Seq Total RNA Pico Input (ZapR Mammalian) SMART-Seq Total RNA Single Cell (ZapR Mammalian) SMART-Seq Stranded Kit* SMARTer Stranded Total RNA-Seq Kit v2 - Pico Input Mammalian*
	Single-cell genome and transcriptome analysis (no UMIs)	Embgenix™ GT-omics Kit

*Product will be phased out soon. Please refer to the product page (via the hyperlink, if available) or contact your local sales representative for more information.

III. Before You Begin

A. Supported Operating Systems

CogentAP is designed to be installed on a server running Linux. The following versions of Linux have been tested and are supported for use with the software:

- CentOS 8 or higher
- RedHat 8 or higher
- Ubuntu 18.04 or higher

B. Hardware Requirements

For analyzing the output of Illumina® NextSeq® High-Output sequencing, the following server requirements (or better) are recommended:

- CPU: 24 cores
- RAM: 64 GB
- Free hard drive space:
  - For all kits apart from the Shasta Total RNA-Seq Kit - 2 Chip and Shasta Whole-Genome Amplification Kit - 2 Chip: at least 1 TB
  - For Shasta Total RNA-Seq Kit - 2 Chip and Shasta Whole-Genome Amplification Kit - 2 Chip: at least 6 times the size of the input FASTQ files (for analyses with default parameters), and 10 times the size of the input FASTQ files (for analyses that will include optional ribodepletion, immune profiling, and fusion analyses)

Testing was also done on MiniSeq™, MiSeq®, HiSeq®, and NovaSeq™ datasets.

- MiniSeq or MiSeq—less computational power may be needed than the specifications described for NextSeq output
- HiSeq or NovaSeq—requires more computational power than described for NextSeq output

Precise hardware requirements were not determined for output from these datasets. Support for performance issues of the servers in conjunction with these dataset types may be limited.

C. User Account Requirements

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

- Where CogentAP will be located,
- Where CogentAP will be run, and
- Where the analyses output will be saved.

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

D. Additional Hardware and Software Dependencies and Recommendations

- **Bash UNIX shell**
- **Internet connectivity on the server**

  The installation process requires internet connectivity, as it sources scripts from GitHub, Bioconda, and CRAN, and downloads genome information from an Amazon S3 bucket. Please ensure that internet connectivity is available on the UNIX server while installing.

- **Conda**

  CogentAP leverages the open-source package manager Conda for installation of CogentAP and its dependencies. Any tools and applications required by CogentAP are installed through Conda inside a local environment explicitly created for CogentAP. Conda installation instructions can be found at https://conda-forge.org/download/.

If Conda is already installed on your server, it is highly recommended to remove the existing installation and install a new version. Instructions for removing an existing Conda installation can be found at https://github.com/conda-forge/miniforge?tab=readme-ov-file#uninstallation and https://docs.anaconda.com/anaconda/uninstall/.

- **bcl2fastq/BCL Convert**

  CogentAP takes raw FASTQ files as input. Sequencer output FASTQ files can be converted to raw FASTQ files using bcl2fastq or bclconvert software from Illumina.

  The bcl2fastq software can be downloaded and installed from https://support.illumina.com/sequencing/sequencing_software/bcl2fastq-conversion-software.html. The bclconvert software can be downloaded and installed from https://support.illumina.com/sequencing/sequencing_software/bcl-convert/downloads.html.

- **Keyboard, monitor, and mouse directly into the server, or a remote access program**

  CogentAP must be run on the Linux server in which it is installed. If users do not have direct console access, a remote access program that enables a Virtual Network Computing (VNC) connection is required through a program such as RealVNC (realvnc.com), TightVNC (tightvnc.com), TigerVNC (tigervnc.org), or similar. Alternatively, users can connect to the Linux server using SSH. In order to keep an SSH-based analysis running when the connection is lost, users can use a screen or tmux session.

  For more information on VNC, along with other VNC clients that can be used, please see the Wikipedia entry at https://en.wikipedia.org/wiki/Virtual_Network_Computing.

E. Required Input Files

- Paired-end read FASTQ files (converted to raw-fastq files using bcl2fastq or BCL Convert)

  NOTE: Single-end FASTQ files cannot be processed using CogentAP

- Sample description file, which can be any one of the following:
  - For experiment results from the Shasta, ICELL8 cx, or ICELL8 systems:
    - Well-list file—a text file output by the Shasta CellSelect® Software that contains well-level sample information. For more information, see Section V.A.6 of the Shasta CellSelect Software User Manual.
  - For plate-based experiment results:
    - Well-list-like format file—a text file that contains sample information, including columns "Barcode" and "Sample". Each column name is case-sensitive. The "Barcode" column contains i7 and i5 indexes concatenated with a plus-sign ("+") (e.g., TAGCGAGT+CCGTTGCG), as in the example below.

```
Sample,Barcode
GM12877,ATGTAAGT+CATAGAGT
GM12877,GCACGGAC+TGCGAGAC
GM12877,GGTACCTT+GACGTCTT
GM05067,AACGTTCC+AGTACTCC
GM06067,GCAGAATT+TGGCCGGT
GM05067,ATGAGGCC+CAATTAAC
```

```
GM08331,AGCCTCAT+TCTCTACT
GM08331,GATTCTGC+CTCTCGTC
GM08331,TCGTAGTG+CCAAGTCT
Etc…
```

For more information about the contents of a well-list-like file, please refer to Shasta Single Cell System User Manual, Appendix C, Section A ("Wells Data Table")

- An Illumina sample sheet—a file format used by Illumina for storing biological sample information and metadata associated with a given experiment.

NOTE: Sample names are used by CogentAP for statistical analysis, such as clustering, and is handled as a group name in the analysis. Illumina sample sheets, natively, require unique sample names for each row in the file, meaning clustering cannot be performed and may cause error messages.

If using an Illumina sample sheet as input that includes multiple experimental instances of one sample, it is recommended that the sample names in the sheet be edited to align with the Well-list-like format usage so more accurate analysis results can be provided.

IV. Software Overview

A. RNA-seq Analysis Workflow

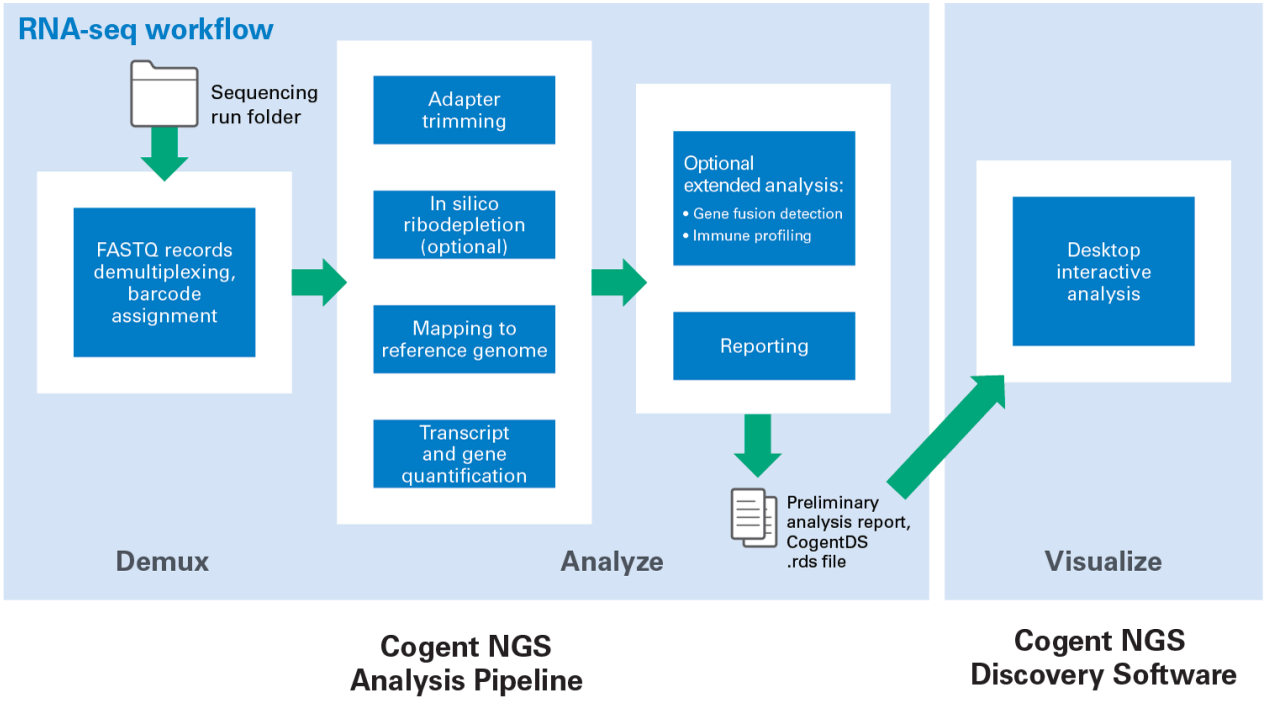


Figure 1. High-level RNA-seq analysis workflow of CogentAP and how its output can be carried over to CogentDS.

For analysis of RNA-seq data, CogentAP consists of two main parts, the demultiplexer (demuxer) and the analyzer.

- The demultiplexer extracts the barcode from the sequencing data (based on the protocol) and writes it into FASTQ files at the end of the read name. There are two options:

- The default splits the data up into barcode-level gzipped FASTQ files, which are required for input into the analyzer.
- The second option leaves the barcode-assigned reads in combined gzip FASTQ files. This format is incompatible with CogentAP v3.1 analysis but can be used with other third-party tools if they support such FASTQ files.

- The analyzer takes the data sent to it by the demultiplexer and performs the following functions:
  - Read trimming (using Cutadapt)
  - Sequencing QC metrics (optional, using FastQC)
  - Genome alignment (using the STAR aligner)
  - Ribodepletion (using SortMeRNA, optional)
  - Deduplication using unique molecular identifiers (UMIs) and unique start stop positions (USSs) (only for reagent kits that employ UMIs, using UMI-tools)
  - Gene expression and transcript expression counting (using Salmon)
  - Gene fusion detection (using STAR-Fusion, optional)
  - Immune profiling analysis (using TRUST4, optional)
  - Summarization (using custom scripts)
  - Generating an HTML report (using a build-in lite version of CogentDS)
- The optional extended analyses (gene fusion and immune profiling analysis) can be launched independently, taking input directly from the analyzer output directory.

NOTE: Not all experiment types support the optional analyses. See Section V.B.2, "Optional Extended Analysis" for more information.

B. DNA-seq Analysis Workflow

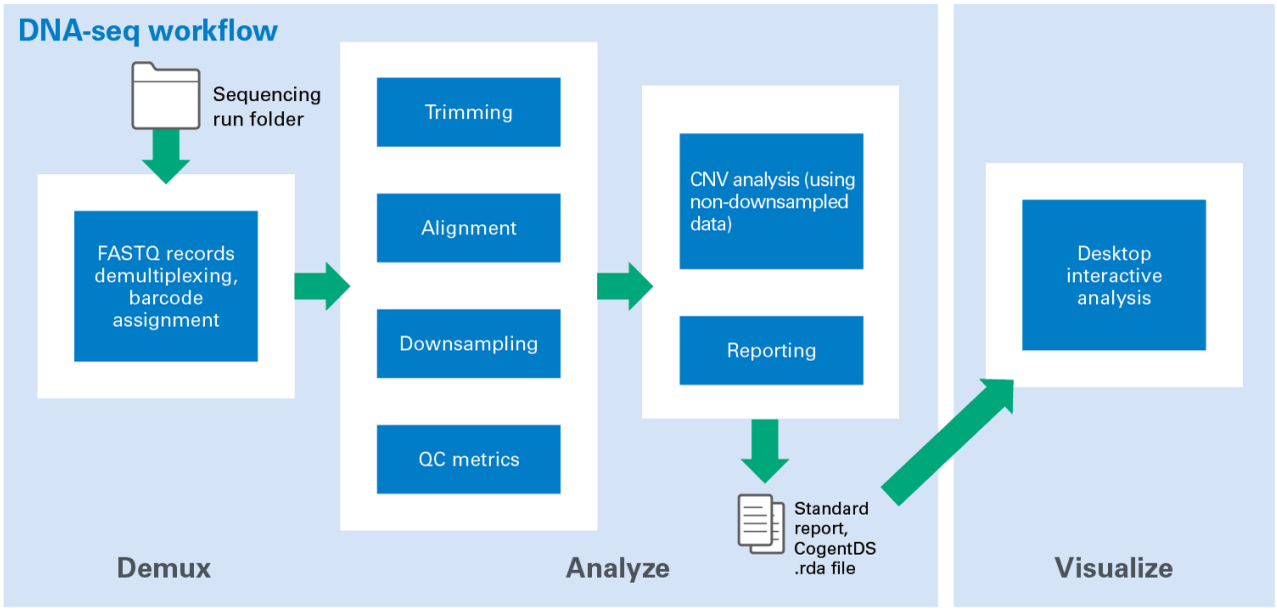


Figure 2. High-level DNA-seq analysis workflow of CogentAP and how its output can be carried over to CogentDS.

CogentAP for DNA-seq analysis also consists of two main parts, the demultiplexer and the analyzer.

- The demultiplexer extracts the barcode from the sequencing data (based on the protocol) and writes it into FASTQ files at the end of the read name. There are two options:
  - The default behavior splits the data up into barcode-level gzipped FASTQ files, which are required for input into the analyzer.
  - The second option leaves the barcode-assigned reads in combined gzip FASTQ files. This format is incompatible with CogentAP v3.1 analysis but can be used with other third-party tools if they support such FASTQ files.
- The analyzer takes the data sent to it by the demultiplexer and performs the following functions:
  - Read trimming (using Trimmomatic)
  - Genome alignment (using the Bowtie2 aligner)
  - Sequencing QC metrics (using FastQC, samtools, Picard, deepTools and MultiQC)
  - Summarization (using custom scripts)
  - CNV calling and generation of CNV-related QC plots (using Ginkgo)
  - Generating QC and Analysis reports (a built-in lite version of CogentDS and custom scripts)

V. Installation & Configuration Options

Run through the steps in this section to set up the Linux server and install CogentAP.

A. Verify the Conda Installation

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

   ```
   conda -V
   ```

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

   e.g.,

   ```
   conda 24.4.0
   ```

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

   ```
   conda activate
   ```

   A successful Conda install will result in a change in the prompt, as shown in Figure 3.

   ```
   $ conda activate
   (base) $
   (base) $ conda dectivate
   $
   ```

   Figure 3. Screenshot of the Linux command line showing a successful check of the base Conda environment. If the Conda installation was not completed as required, both commands would return error messages.

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

```
conda deactivate
```

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

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

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

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

      ```
      more ~/.bash_profile
      ```

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

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

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

      e.g., username is 'myacct':

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

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

B. Uninstall Previous Instances of CogentAP

NOTE: If CogentAP has never been installed on the server, skip to the next section (Section IV.C).

If an earlier version of CogentAP was installed on the server, it should be uninstalled prior to installing Cogent NGS Analysis Pipeline v3.1.

Follow the uninstall directions in Section IV.E ("How to Uninstall CogentAP").

C. Install Cogent NGS Analysis Pipeline v3.1

CogentAP is available for download as a compressed file from the CogentAP product page.

1. Download the installation ZIP file (`Cogent_NGS_Analysis_Pipeline_v3.1.zip`), following the directions (a) on the page seen after submitting the sign-up form on the CogentAP product page or (b) in the confirmation email sent to the email address submitted in the form.
2. Move or copy the CogentAP ZIP file onto the Linux server into the directory location where you want to install CogentAP.

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

3. From the same directory location in Step 2, run the following two commands in the order listed:

   ```
   unzip Cogent_NGS_Analysis_Pipeline_v3.1.zip && \
   mv Cogent_NGS_Analysis_Pipeline_v3.1 CogentAP
   ```

   ```
   cd CogentAP
   ```

   The `CogentAP` directory contains files and directories required by the pipeline's scripts.

```
.
├── bin
├── cogent
├── cogent_ap_env.yaml
├── CogentAP_setup.sh
├── CogentAP_tools
├── cogent_cli.py
├── config
├── deploy
├── lib
├── main.nf
├── modules
├── nextflow.config
├── pyproject.toml
├── README.md
├── run_tests.sh
├── test
├── VERSION
└── workflows
```

Figure 4. The sub-directory and files list of the `CogentAP` folder.

4. Run the following command to install CogentAP and its dependencies:

   ```
   bash CogentAP_setup.sh install
   ```

   Once the installation is complete, the following message will display.

```
Successfully installed CogentAP pipeline and dependencies.
Please setup genomes next. Pre-indexed genomes can be downloaded by running "bash CogentAP_setup.sh genome_install ${NAME}", where ${NAME} is hg38 or mm39.
```

Figure 5. Console message illustrating a successful CogentAP software install on the Linux server.

5. Run the following command to install the human genome build:

   ```
   bash CogentAP_setup.sh genome_install hg38
   ```

   (Optional): If you will be analyzing sequence data for *Mus musculus* (mice), run the following command to install the mouse genome build:

   ```
   bash CogentAP_setup.sh genome_install mm39
   ```

   NOTE: Each genome installation process will take approximately 1 hr to complete, depending on the computational capacity of the server and the download speed of the internet connection.

If the genome is successfully installed, a message with the text "Successfully installed genome" will be displayed. After a genome build is installed, CogentAP is ready to use.

D. (Optional) Set Up `$COGENT_AP_HOME` Environmental Variable

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

Example:

If your account name is 'myacct', the absolute pathname for myacct's home directory is `/home/myacct`, and CogentAP was installed in the `~/bin` directory, edit `.bash_profile` to add the following line:

```
export COGENT_AP_HOME=/home/myacct/bin/CogentAP
```

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

```
source ~/.bash_profile
```

The phrase `$COGENT_AP_HOME` can then be used as an alias shortcut to reference `/home/myacct/bin/CogentAP`.

Example:

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

```
cd $COGENT_AP_HOME
```

NOTE: Subsequent references to `$COGENT_AP_HOME` in this document refer to the full path where the CogentAP software is installed.

E. How to Uninstall CogentAP

CogentAP can be uninstalled by deleting the `CogentAP/` software directory from the server.

If `$COGENT_AP_HOME` has been defined in `.bash_profile`, edit the file to remove the reference to `$COGENT_AP_HOME` as well.

NOTE:

- If you've stored output files from previous analysis runs in the `CogentAP/` directory that you would like to save, make sure to move them out of the directory prior to deleting it.
- If you used an older version of this software, called mappa™ Analysis Pipeline, delete the entire `mappa/` directory to uninstall it.

VI. Running the Pipeline

Before running an analysis, raw-fastq files need to be generated from the sequencer-output FASTQ files. Once the raw-fastq files are created, CogentAP can be run using the appropriate experiment type.

Tables 2 and 3 map the experiment type and corresponding kits (from Table 1) to the protocol option names/abbreviations used to select them; Table 2 is for Shasta and ICELL8 cx applications, while Table 3 lists plate-based applications.

Table 2. Shasta and ICELL cx experiment type options for the validated Takara Bio reagent kits.

Experiment type and supported kits	Experiment Type
Shasta, ICELL8 cx, and ICELL8 single-cell full gene-body transcriptome analysis • SMART-Seq Pro Application Kit - 2 Chip	`icell8_fla`
Shasta and ICELL8 cx single-cell full gene-body total RNA-seq analysis • Shasta Total RNA-Seq Kit - 2 Chip	`shasta_total_rna`
Shasta and ICELL8 cx single-cell whole-genome amplification analysis	`shasta_wga`

• Shasta Whole-Genome Amplification Kit – 2 Chip	

Table 3. Plate-based experiment type options for the validated Takara Bio reagent kits.

Experiment type and supported kits	Experiment Type
Plate-based full gene-body transcriptome analysis with UMIs • SMART-Seq mRNA LP (with UMIs)	`smartseq_fla_umi`
Plate-based full gene-body transcriptome analysis • SMART-Seq mRNA LP • SMART-Seq mRNA • SMART-Seq mRNA Single Cell LP • SMART-Seq mRNA Single Cell • SMART-Seq mRNA HT LP • SMART-Seq mRNA HT • SMART-Seq v4 PLUS Kit • SMART-Seq v4 Ultra Low Input RNA Kit • SMART-Seq Single Cell PLUS Kit • SMART-Seq Single Cell Kit • SMART-Seq HT PLUS Kit • SMART-Seq HT	`smartseq_fla`
Plate-based strand-specific total RNA-seq for mammalian samples (with UMIs) • SMART-Seq Total RNA Pico Input with UMIs (ZapR Mammalian) • SMARTer Stranded Total RNA-Seq Kit v3 - Pico Input Mammalian	`stranded_umi`
Plate-based strand-specific RNA-seq (no UMIs) • SMART-Seq Total RNA Pico Input (ZapR Mammalian) • SMART-Seq Total RNA Single Cell (ZapR Mammalian) • SMART-Seq Stranded Kit	`stranded`

- SMARTer Stranded Total RNA-Seq Kit v2 - Pico Input Mammalian

Plate-based single-cell genome and transcriptome analysis (no UMIs) • Embgenix GT-omics Kit	`shasta_wga` (genome) `smartseq_fla` (transcriptome)

Optional extended RNA-seq analysis (Section V.B.2)

Gene fusion detection and immune profiling analysis are explicitly included to leverage our full gene-body chemistry advantages over the other 3′ chemistries; refer to Section V.B.2 for how to run these extended analyses. While we provide these analyses for our other chemistries, we do not recommend it for them.

For an overview of analysis types available in CogentAP for the Takara Bio chemistries supported by the software, refer to the table on the bioinformatics portal on our website.

A. Generation of raw-fastq Files

The CogentAP demultiplexer takes one pair of raw-fastq files as input (i.e., not split by barcode). The following procedure converts the sequencer FASTQ output files into the format expected by CogentAP using bcl2fastq or BCL Convert.

1. Log in to the server that stores the sequencing run output folder. This server will typically have bcl2fastq or bclconvert installed (see Section II.D for more information about bcl2fastq and bclconvert).
2. Change to the directory where you want the raw-fastq files to be created.
3. From the server where CogentAP is installed, copy the `SampleSheet_dummy_bcl2fastq.csv` file, located in the `$COGENT_AP_HOME/config` folder, into the directory selected in Step 2.
4. Run bcl2fastq or BCL Convert:

   If running bcl2fastq, use the following syntax:

```
bcl2fastq -R <RUN_FOLDER> \
  -o <RUN_ID> \
  --no-lane-splitting \
  --sample-sheet SampleSheet_dummy_bcl2fastq.csv <RUN_ID>.stdout \
  2 > <RUN_ID>.stderr
```

   where:

   - `<RUN_FOLDER>` is the path to the sequencing run folder and
   - `<RUN_ID>` is the ID automatically generated by Illumina sequencer

   NOTE: Some versions of bcl2fastq have a bug where the indexes required for demultiplexing will not be inserted into the raw-fastq if a sample sheet file is not specified in the command syntax. To prevent encountering the issue, we recommend always using the `SampleSheet_dummy_bcl2fastq.csv` option when generating the raw-fastq files from the bcl2fastq command.

   If running BCL Convert, use the following syntax template:

```
bcl-convert --bcl-input-directory <RUN_FOLDER> \
  --output-directory <RUN_ID> --no-lane-splitting \
  --sample-sheet=DummySampleSheet \
  > <RUN_ID>.stdout 2 > <RUN_ID>.stderr
```

where:

- <RUN_FOLDER> is the path to the sequencing run folder and
- <RUN_ID> is the ID automatically generated by Illumina sequencer

NOTE: Templates for the DummySampleSheet for BCL Convert can be found in $COGENT_AP_HOME/config. Modify the read and index lengths as necessary based on the documentation for BCL Convert from Illumina's website.

5. Retrieve the raw-fastq files from the <RUN_ID> folder located in the working directory from Step 2. These are typically named in the syntax Undetermined_*.fastq.gz.

NOTE: To reduce downstream processing time, we recommend that the raw-fastq files are moved to a directory on the server where CogentAP is installed.

B. RNA-Seq Analysis

1. Primary Analysis Commands

NOTE: See Section VI for an example of the full syntax for the command-line scripts.

For RNA-seq analysis, CogentAP starts from the main script, cogent, and has defined subcommands, listed below.

Demux and Analyze

- rna demux
- rna analyze

Optional Extended Analysis

- rna postprocess immune
- rna postprocess fusion

Additional Commands

- rna add_genome

These scripts can be launched from any location (working directory) on the Linux server where the CogentAP software is installed. The full list of arguments can be accessed using the syntax:

```
$COGENT_AP_HOME/cogent <COMMAND> -h
```

The rna demux and rna analyze commands are the core functionality of the RNA-seq analysis and are described below. Section V.B.2 covers the extended analysis options, while Section V.B.3 describes the additional available commands.

NOTE: If analyzing data from the Shasta Total RNA-Seq Kit, read counts for all the barcodes in the experiment must be estimated before running the demux process. This process is outlined in "RNA Demux and Dry Run (for analysis of Shasta Total RNA-Seq Kit data)", below.

a) RNA Demux (for analysis of data from all kits except the Shasta Total RNA-Seq Kit)

In general, the demuxer (`cogent rna demux`) is run first to generate demultiplexed FASTQ files.

The full list of `rna demux` arguments are listed in Table 4 and a screenshot of the output `$COGENT_AP_HOME/cogent rna demux -h` is shown in Figure 6.

Table 4. Full list of options under `cogent rna demux -h`.

Option	Description	Default
`-h, --help`	Produces a help message.	N/A
`-f, --fastq1`	Specifies the input Read1 (R1) FASTQ file.	N/A
`-p, --fastq2`	Specifies the input Read2 (R2) FASTQ file.	N/A
`-t {}, --type_of_experiment {}`	Specifies the experimental protocol (See Tables 2 and 3 for experimental protocols based on reagent kit).	N/A
`-o, --output_dir`	Indicates the path to the application output.	N/A
`-b, --barcodes-file`	Specifies path to the well-list file from CellSelect software or another custom file containing only barcodes that were selected for sequencing.	N/A
`--fastqc`	Runs FASTQC to create quality reports for FASTQ files.	disabled
`-m {}, --mismatch{}`	Specifies the number of allowed mismatched bases per barcode.	1
`-u, --umi_length`	Overwrites the UMI length associated with the experimental type.	Calculated based on experiment type
`-n, --n_processes`	Specifies the number of demultiplexing processes to spawn during execution. The maximum value (*N*) should not exceed the number of CPUs on the server.	15 (Fixed at 3 when `--no_split_fastqs` is used)
`--n_writers`	Specifies the number of demultiplexing writing processes to spawn during execution. The maximum value should be less than or equal to N-2.	8 (Fixed at 1 when `--no_split_fastqs` is used)
`--no_gz`	Specifies not to compress (gzip) output FASTQ files	FASTQ files are compressed
`--undetermined_fq`	Saves undetermined/unselected/short reads to an undetermined FASTQ file	Reads are not saved

Option	Description	Default
`--i7_rc {auto, true, false}`	Reverse-complement I7 Index. Default of "auto" detects and auto-corrects the reverse complement of I7 indices by certain Illumina sequencers. Otherwise, manually override with "true" or "false"*.	auto
`--i5_rc {auto, true, false}`	Reverse-complement I5 Index. Enter "auto" to detect and auto-correct the reverse complement of I5 indices by certain Illumina sequencers. Otherwise, manually override with "true" or "false"*.	auto
`--read_buffer`	Specifies buffer size of the data sent to each demultiplexing process in GB.	0.1
`--prog`	Specifies number of reads to process before updating in the log file	10,000,000
`--no_split_fastqs`	Output merged FASTQ file(s). Barcodes are written into read names and merged into a single pair of large FASTQ files.	disabled
`--use_barcodes`	Limits the number of barcodes to a specified value	10,000
`--check_reads`	Uses a specified number of reads to estimate read counts during barcode selection.	200,000,000
`--min_reads`	Discards barcodes with estimated read count lower than this number	1
`--random_pick`	Picks random reads during barcode selection	disabled
`--dry_run`	Outputs estimated counts of all barcodes without writing demultiplexed FASTQ files	disabled
`--preview`	Prints the Nextflow command without executing it to verify parameters before starting the pipeline.	N/A

*Manually specify “true” or “false’ when you definitively know the orientation of I7 or I5 indexes in the well-list file. “True” automatically treats indexes as reverse-complement and corrects all indexes in the well-list file.

```
usage: cogent rna demux [-h] -f FASTQ1 -p FASTQ2 -t
                        {stranded,stranded_umi,smartseq_fla,smartseq_fla_umi,icell8_fla,smartseq_pro,shasta_total_rna}
                        -o OUTPUT_DIR -b BARCODES_FILE [--fastqc] [-m {0,1}] [-u UMI_LENGTH] [-n N_PROCESSES]
                        [--n_writers N_WRITERS] [--no_gz] [--undetermined_fq] [--i7_rc {auto,true,false}]
                        [--i5_rc {auto,true,false}] [--read_buffer READ_BUFFER] [--prog PROG]
                        [--no_split_fastqs] [--use_barcodes USE_BARCODES] [--check_reads CHECK_READS]
                        [--min_reads MIN_READS] [--random_pick] [--dry_run] [--preview]

            Script to de-multiplex barcoded reads from sequence data stored in
            FASTQ files. User options are designed to simplify de-multiplexing
            for experiments derived from Takara protocols. Barcode (and optionally
            UMI) sequences are extracted and stored in the read name. Users may
            specify whether the resulting de-multiplexed data are merged, or split
            into individual barcode-level files.

options:
  -h, --help            show this help message and exit
  -f FASTQ1, --fastq1 FASTQ1
                        Input Read1 (R1) FASTQ file.
  -p FASTQ2, --fastq2 FASTQ2
                        Input Read2 (R2) FASTQ file.
  -t {stranded,stranded_umi,smartseq_fla,smartseq_fla_umi,icell8_fla,smartseq_pro,shasta_total_rna}, --type_of_experiment {strande
d,stranded_umi,smartseq_fla,smartseq_fla_umi,icell8_fla,smartseq_pro,shasta_total_rna}
                        Experimental protocol used.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Name of output directory to store results.
  -b BARCODES_FILE, --barcodes_file BARCODES_FILE
                        Well List file from Takara's CellSelect Software (Recommended), or another custom file
                        containing only barcodes that were selected for sequencing.
  --fastqc              Run FASTQC to create quality reports for FASTQ files.
  -m {0,1}, --mismatch {0,1}
                        Number of allowed mismatched bases per barcode.
  -u UMI_LENGTH, --umi_length UMI_LENGTH
                        Overwrite UMI length. By default, length is automatically determined by experiment
                        type.
  -n N_PROCESSES, --n_processes N_PROCESSES
                        Number of demultiplexing processes to spawn during execution.
  --n_writers N_WRITERS
                        Number of demultiplexing writing processes to spawn during execution.
  --no_gz               Do not compress (gzip) output FASTQ files.
  --undetermined_fq     Save Undetermined/Unselected/Short reads to Undetermined FASTQ files.
  --i7_rc {auto,true,false}
                        Reverse-complement I7 Index (Full Length protocol only). Enter "auto" to detect and
                        auto-correct the reverse complementation of I5/I7 indices by certain Illumina
                        sequencers. Otherwise manually override with "True" or "False").
  --i5_rc {auto,true,false}
                        See help section for "--i7_rc".
  --read_buffer READ_BUFFER
                        Buffer size of data sent to each demultiplexing (worker) process in GB.
  --prog PROG           Number of reads to process before updating status in log file.
  --no_split_fastqs     Output merged FASTQ file(s). Barcodes are written into read names and merged into large
                        FASTQ file. By default output into barcode-level FASTQ files.
  --use_barcodes USE_BARCODES
                        Limit number of barcodes to this value.
  --check_reads CHECK_READS
                        Use this number of reads to estimate read counts during barcode selection.
  --min_reads MIN_READS
                        Discard barcodes with estimated read count lower than this number.
  --random_pick         Pick random reads during barcode selection, rather than anazlyzing the first N read
                        pairs.
  --dry_run             Output estimated counts of all barcodes, but do not write demultiplexed FASTQs.
  --preview             Print nextflow command without executing it.
```

Figure 6. The output of `cogent rna demux -h` at the command line.

b) RNA Demux and Dry Run (for analysis of Shasta Total RNA-Seq Kit data)

For data derived from a Shasta Total RNA-Seq Kit, read counts for all the barcodes in the experiment must be estimated before running the demux process. To perform the estimation, demultiplex using the `--dry_run` mode.

```
$COGENT_AP_HOME/cogent rna demux \
  --dry_run \
```

```
  -f <FASTQ_R1> \
  -p <FASTQ_R2> \
  -b <WELLLIST> \
  -t shasta_total_rna \
  -o <OUTPUT>
```

where:

- <FASTQ_R1> and <FASTQ_R2> are the full paths to the FASTQ files generated by an Illumina sequencing platform.
- <WELLLIST> is the full path to the Shasta system WellList, located at $COGENT_AP_HOME/config/well_list_shasta_total_rna.csv.
- <OUTPUT> is a string; it will be the name of the output folder created by the analysis.

The estimated counts for each barcode can be found in the path listed below and can be imported into CogentDS to generate a knee plot for determining the optimal and/or minimum number of reads required to retain a barcode for downstream analysis:

<OUTPUT>/demultiplexed_fastqs/demultiplexed_fastqs_counts_all.estimated.csv

To demultiplex (demux), set the number of barcodes to keep (--use_barcodes) and/or the minimum number of reads required (--min_reads) to keep a barcode, based on the estimated counts of all barcodes from the demux dry run. The more stringent of these two parameters will be applied as the filter to select the barcodes in the demux process.

```
$COGENT_AP_HOME/cogent rna demux \
  --use_barcodes <NUMBER OF BARCODES TO KEEP> \
  --min_reads <MINIMUM NUMBER OF READS REQUIRED TO KEEP A BARCODE> \
  -f <FASTQ_R1> \
  -p <FASTQ_R2> \
  -b <WELLLIST> \
  -t shasta_total_rna \
  -o <OUTPUT>
```

RNA Analyze

The resulting directory of FASTQ files from `cogent rna demux` is used as input to run the analyzer (`cogent rna analyze`) to obtain the output files described in Section VII.

The full list of `rna analyze` control options are listed in Table 5 and a screenshot of the output of `$COGENT_AP_HOME/cogent rna demux -h` is shown in Figure 7. For an example on how to run analysis on the test dataset, see Section VI.

Table 5. Full list of options under `cogent rna analyze -h` .

Option	Description	Default
`-h, --help`	Produces a help message.	N/A
`-g {}, --genome {}`	Allows for selection of a supported genome or custom genome that you have installed	N/A

`-o, --output_dir`	Specifies the output directory in which to store the results of the pipeline	N/A
`-i, --input_dir`	Specifies the input directory that contains the results from `rna demux`	N/A
`-G, --genome_dir`	Specifies the directory where the genome and index files are installed	`$COGENT_AP_HOME/genomes`
`--fastqc`	Runs FASTQC to create quality reports for FASTQ files	disabled
`-t {}, --type_of_experiment {}`	Specifies experimental protocol to be used	N/A
`--immune`	Generates immune profiling matrix	disabled
`--fusion`	Generates gene fusion matrix	disabled
`--ribodeleption {auto, true,false}`	Removes ribosomal RNA reads	auto
`--keep_intermediate`	Saves intermediate files from analysis such as BAM files	disabled
`--resume`	Resumes a previous pipeline run with the same inputs.	N/A
`--preview`	Prints the Nextflow command without executing it to verify parameters before starting the pipeline.	N/A

```
usage: cogent rna analyze [-h] -g {hg38,mm39} [-G GENOME_DIR] -o OUTPUT_DIR -i INPUT_DIR [--fastqc] -t
                          {stranded,stranded_umi,smartseq_fla,smartseq_fla_umi,icell8_fla,smartseq_pro,shasta_tot
al_rna}
                          [--immune] [--fusion] [--ribodepletion {auto,true,false}] [--keep_intermediate]
                          [--preview] [--resume]

            Script to perform counting analysis for exons and genes by fastq input data.
            The input to this script are files output by Cogent demux.
            The fastq files are expected to contain the barcode info in the read name.
            Optionally it can also contain UMI info following the BC.
            The modules currently included are:
                - Trimming (cutadapt)
                - Alignment (STAR)
                - Counting (featureCounts)
                - Summarization (TBUSA)
                - Reporting (TBUSA, CogentDS)

options:
  -h, --help            show this help message and exit
  -g {hg38,mm39}, --genome {hg38,mm39}
                        Select a supported genome or provide the name of a custom genome that you installed.
  -G GENOME_DIR, --genome_dir GENOME_DIR
                        Directory where genome and index files were installed by add_genome. [Default:
                        $COGENT_ROOT/genomes]
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Name of output directory to store results.
  -i INPUT_DIR, --input_dir INPUT_DIR
                        Directory contains results from demux command. The directory must contain FASTQ files
                        after demultiplexing.
  --fastqc              Run FASTQC to create quality reports for FASTQ files.
  -t {stranded,stranded_umi,smartseq_fla,smartseq_fla_umi,icell8_fla,smartseq_pro,shasta_total_rna}, --type_of_ex
periment {stranded,stranded_umi,smartseq_fla,smartseq_fla_umi,icell8_fla,smartseq_pro,shasta_total_rna}
                        Experimental protocol used.
  --immune              Generate immune profiling matrix.
  --fusion              Generate gene fusion matrix.
  --ribodepletion {auto,true,false}
                        Counting and removal of ribosomal RNA. Enter "auto" for depletion based on kit
                        type.Otherwise manually override with "true" or "false".
  --keep_intermediate   Save genome and transcriptome BAM files from STAR.
  --preview             Print nextflow command without executing it.
  --resume              Resume a previous pipeline run with the same inputs
```

Figure 7. The output of `cogent rna analyze -h` at the command line.

NOTE: The `--ribodepletion` parameter is set to 'auto' by default. 'auto' mode enables in silico ribodepletion by default on all RNA kits except for the Shasta Total RNA-Seq Kit, for which it is disabled by default.

To enable ribodepletion during Shasta Total RNA-Seq Kit analysis, the `--ribodepletion` parameter has to be set to 'true'. To disable ribodepletion in other kits, `--ribodepletion` can be set to 'false'.

2. Optional Extended Analysis

a) Gene Fusion Analysis

Gene fusion analysis is launched by the command

```
$COGENT_AP_HOME/cogent rna postprocess fusion
```

You can also launch this as an option while running the analyzer with the option `--fusion`

```
$COGENT_AP_HOME/cogent rna analyze --fusion
```

to launch gene fusion analysis at the same time. The options for gene fusion analysis can be viewed with the option `-h` (Figure 8).

```
$COGENT_AP_HOME/cogent rna postprocess fusion -h
```

The resulting `CogentDS_analysis.rds` file includes gene fusion detection and all other analysis done with the `rna analyze` command.

NOTE: The Rdata file from gene fusion analysis is only generated when run as an option when running the pipeline. When launched as a standalone gene fusion analysis, the resulting output files (in mtx format) cannot currently be used for downstream analysis with CogentAP or CogentDS.

```
usage: cogent rna postprocess fusion [-h] -i INPUT_DIR -o OUTPUT_DIR -g {hg38,mm39} [-G GENOME_DIR] [--resume] [--preview]

       A command to perform gene fusion detection analysis.
       The input to this command is result directory from analyze command.
       The directory is expected to contain junction information files (.Chimeric.out.junction) and stats.csv
       This analysis ignores UMI even if UMI enabled experiment type is specified.

options:
  -h, --help            show this help message and exit
  -i INPUT_DIR, --input_dir INPUT_DIR
                        Directory contains results from analyze command. The directory must contain genematrix and *_stats.csv.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Name of output directory to store results.
  -g {hg38,mm39}, --genome {hg38,mm39}
                        Select a supported genome or provide the name of a custom genome that you installed.
  -G GENOME_DIR, --genome_dir GENOME_DIR
                        Directory where genome and index files were installed by add_genome. [Default: $COGENT_ROOT/genomes]
  --resume              Resume a previous pipeline run with the same inputs
  --preview             Print nextflow command without executing it.
```

Figure 8. The output of `cogent rna postprocess fusion -h` at the command line.

b) *Immune Profiling Analysis*

Immune profiling analysis is launched by the command

```
$COGENT_AP_HOME/cogent rna postprocess immune
```

You can also launch this as an option while running the analyzer with the option `--immune`

```
$COGENT_AP_HOME/cogent analyze --immune
```

to launch immune profiling analysis at the same time. The options immune profiling analysis can be viewed with the option `-h` (Figure 9).

```
$COGENT_AP_HOME/cogent rna postprocess immune -h
```

The resulting `CogentDS_analysis.rds` file includes detected clonotypes and all other analysis done with the `analyze` command.

NOTE: The Rdata file from immune profiling analysis is only generated when run as an option when running the analyzer. When launched as a standalone immune analysis, the resulting output files cannot be currently used for downstream analysis with CogentAP or CogentDS.

```
usage: cogent rna postprocess immune [-h] -i INPUT_DIR -o OUTPUT_DIR -g {hg38,mm39} [-G GENOME_DIR] [--resume] [--preview]

       A command for perfoming immune-profiling on split fastqs.
       The input to this script are files output by Cogent demux.
       The fastq files are expected to contain the barcode info in the read name.
       The modules currently included are:
           -- read assembly & clonotype identification (Trust4)

options:
  -h, --help            show this help message and exit
  -i INPUT_DIR, --input_dir INPUT_DIR
                        Directory contains results from analyze command. The directory must contain genematrix and *_stats.csv.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Name of output directory to store results.
  -g {hg38,mm39}, --genome {hg38,mm39}
                        Select a supported genome or provide the name of a custom genome that you installed.
  -G GENOME_DIR, --genome_dir GENOME_DIR
                        Directory where genome and index files were installed by add_genome. [Default: $COGENT_ROOT/genomes]
  --resume              Resume a previous pipeline run with the same inputs
  --preview             Print nextflow command without executing it.
```

Figure 9. The output of `cogent rna postprocess immune -h` at the command line.

3. Adding a Genome Build

The human and mouse genome builds available from our server (Section IV.C, "Install Cogent NGS Analysis Pipeline v3.1) are recommended for use in the pipeline, but genomes of other species can be added into the software post-install.

NOTE: Extended analysis for gene fusion detection or immune profiling is not supported for custom genome builds added through this process.

To add custom genome data to CogentAP:

1. Create a copy of the file under

   `$COGENT_AP_HOME/config/genome_sources/sample.config`

   and rename it

   `$COGENT_AP_HOME/config/genome_sources/`**`<common_species_name>`**`.config`

   where `<common_species_name>` is the name of the genome being added (e.g., `dm6`)

2. Update the following fields using a text editor:

   - Replace 'GENOME' with the value of `<common_species_name>` from Step 1 (e.g., `dm6`).
   - Replace 'ENSEMBL_GENOME_FASTA_URL' with the public URL of the FASTA file containing all the sequences (chromosomes and contigs) from Ensembl.

     Using the fruit fly genome from Ensembl.org as an example, you would replace 'ENSEMBL_GENOME_FASTA_URL' with the following URL:

     ```
     https://ftp.ensembl.org/pub/release-
     113/fasta/drosophila_melanogaster/dna/Drosophila_melanogaster.
     BDGP6.46.dna_sm.toplevel.fa.gz
     ```

- Replace 'ENSEMBL_GTF_URL' with the public URL of the GTF file containing the annotation and, importantly, the gene information for analysis from Ensembl.

  Using the fruit fly genome from Ensembl.org as an example, you would replace 'ENSEMBL_GTF_URL' with the following URL:

  ```
  https://ftp.ensembl.org/pub/release-113/gtf/drosophila_melanogaster/Drosophila_melanogaster.BDGP6.46.113.gtf.gz
  ```

- Replace 'PATH_TO_SORTMERNA_FASTAS' with the location of the file (path) of the FASTA files that contain ribosomal sequences to be used for ribodepletion.
- Replace 'PATH_TO_MITO_GENES' with the location of the file (path) containing a list of mitochondrial genes in ENSEMBL format, one gene listed per line.

  NOTE: As FASTA and GTF files are a standard file format, files from any source should work. However, the pipeline has only been tested on genomes downloaded from Ensembl. If a problem is encountered using files from another source, it is recommended to try importing a genome using files from Ensembl.org. For RNA-seq analysis, the line 'ginkgo_ref' in the config file can be safely removed.

3. Run the script:

   ```
   $COGENT_AP_HOME/cogent rna add_genome \
     -g <common_species_name> \
     -G <genome_dir>
   ```

   where `<common_species_name>` is the name of the genome being added, and `<genome_dir>` is the directory where the new genome will be stored. If `-G` is not used, the genome is stored by default in `$COGENT_AP_HOME/genomes`.

   For additional help with this script, type:

   ```
   $COGENT_AP_HOME/cogent rna add_genome -h
   ```

CogentAP should now be able to analyze data with the genome option `-g` set to `<common_species_name>` during `cogent rna analyze`.

4. Processing Time

The time taken by the pipeline will vary based on the hardware specifications of the server on which it is run, the size of the raw-fastq input files, and where the files are stored.

During testing, a combined demultiplexing and analysis run for data generated by MiSeq (~25M read pairs) against raw-fastq files stored locally (on the same server CogentAP was installed) typically took about 1–1.5 hr to process. A NextSeq High Output run (~400M read pairs) from local raw-fastq files typically took ~10–12 hr to complete. Input taken from a NovaSeq run (~2G read pairs, or more) will take even longer. Data generated with the Shasta Total RNA-Seq Kit could take 48–60 hr to complete.

If the raw-fastq files are instead stored on a network drive, these baselines might be exceeded.

C. DNA-Seq Analysis

1. Primary Analysis Commands

For DNA-seq or Shasta WGA analysis, CogentAP starts from the main script, `cogent`, and has defined subcommands, listed below.

Demux and Analyze

- `dna demux`
- `dna analyze`

Additional Commands

- `dna add_genome`

These scripts can be launched from any location (working directory) on the Linux server where the CogentAP software is installed. The full list of options can be accessed using the syntax:

```
$COGENT_AP_HOME/cogent <COMMAND> -h
```

The `dna demux` and `dna analyze` commands are described below.

DNA Demux and DNA Analyze

For DNA-seq or Shasta WGA analysis, the demuxer (`cogent dna demux`) is run first to generate demultiplexed FASTQ files. The resulting directory of FASTQ files is then used as input to run the analyzer (`cogent dna analyze`) to obtain the output files described in Section VII.

The full list of `dna demux` arguments are listed in Table 6 and a screenshot of the output of `$COGENT_AP_HOME/cogent dna demux -h` is shown in Figure 10.

Table 6. Full list of options under `cogent dna demux -h`.

Option	Description	Default
`-h, --help`	Produces a help message.	N/A
`-f, --fastq1`	Specifies the input Read1 (R1) FASTQ file.	N/A
`-p, --fastq2`	Specifies the input Read2 (R2) FASTQ file.	N/A
`-t {}, --type_of_experiment {}`	Specifies experimental protocol to be used (see Tables 2 and 3).	N/A
`-o, --output_dir`	Indicates the path to the application output	N/A
`-b, --barcodes-file`	Specifies path to the well-list file from CellSelect Software or another custom file containing only barcodes that were selected for sequencing.	N/A
`--fastqc`	Runs FASTQC to create quality reports for FASTQ files.	disabled
`-m {}, --mismatch{}`	Specifies the number of allowed mismatched bases per barcode.	1

Option	Description	Default
-n, --n_processes	Specifies the number of demultiplexing processes to spawn during execution. The maximum value (N) should not exceed the number of CPUs on the server.	15 (Fixed at 3 when --no_split_fastqs is used)
--n_writers	Specifies the number of demultiplexing writing processes to spawn during execution. The maximum value should be less than or equal to N-2.	8 (Fixed at 1 when --no_split_fastqs is used)
--no_gz	Do not compress (gzip) output FASTQ files	FASTQ files are compressed
--undetermined_fq	Save undetermined/unselected/short reads to an undetermined FASTQ file	Reads are not saved
--i7_rc {auto, true, false}	Reverse-complement I7 Index. Default of "auto" detects and auto-corrects the reverse complement of I7 indices by certain Illumina sequencers. Otherwise, manually override with "true" or "false"*.	auto
--i5_rc {auto, true, false}	Reverse-complement I5 Index. Default of "auto" detects and auto-corrects the reverse complement of I7 indices by certain Illumina sequencers. Otherwise, manually override with "true" or "false"*.	auto
--read_buffer	Specifies buffer size of the data sent to each demultiplexing process in GB.	0.1
--prog	Specifies the number of reads to process before updating in the log file.	10,000,000
--no_split_fastqs	Output merged FASTQ file(s). Barcodes are written into read names and merged into a single pair of large FASTQ files.	disabled
--use_barcodes	Limit the number of barcodes to this number.	10,000
--check_reads	Use this number of reads to estimate read counts during barcode selection.	200,000,000
--min_reads	Discards barcodes with estimated read count lower than this number.	1

Option	Description	Default
`--random_pick`	Picks random reads during barcode selection rather than analyzing the first N read pairs.	disabled
`--preview`	Prints the Nextflow command without executing it to verify parameters before starting the pipeline.	N/A

```
usage: cogent dna demux [-h] -f FASTQ1 -p FASTQ2 -t {shasta_wga} -o OUTPUT_DIR -b BARCODES_FILE [--fastqc] [-m {0,1}]
                        [-n N_PROCESSES] [--n_writers N_WRITERS] [--no_gz] [--undetermined_fq] [--i7_rc {auto,true,false}]
                        [--i5_rc {auto,true,false}] [--read_buffer READ_BUFFER] [--prog PROG] [--no_split_fastqs]
                        [--use_barcodes USE_BARCODES] [--check_reads CHECK_READS] [--min_reads MIN_READS] [--random_pick]
                        [--preview]

            Script to de-multiplex barcoded reads from sequence data stored in
            FASTQ files. User options are designed to simplify de-multiplexing
            for experiments derived from Takara protocols. Barcode (and optionally
            UMI) sequences are extracted and stored in the read name. Users may
            specify whether the resulting de-multiplexed data are merged, or split
            into individual barcode-level files.

options:
  -h, --help            show this help message and exit
  -f FASTQ1, --fastq1 FASTQ1
                        Input Read1 (R1) FASTQ file.
  -p FASTQ2, --fastq2 FASTQ2
                        Input Read2 (R2) FASTQ file.
  -t {shasta_wga}, --type_of_experiment {shasta_wga}
                        Experimental protocol used.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Name of output directory to store results.
  -b BARCODES_FILE, --barcodes_file BARCODES_FILE
                        Well List file from Takara's CellSelect Software (Recommended), or another custom file containing only
                        barcodes that were selected for sequencing.
  --fastqc              Run FASTQC to create quality reports for FASTQ files.
  -m {0,1}, --mismatch {0,1}
                        Number of allowed mismatched bases per barcode.
  -n N_PROCESSES, --n_processes N_PROCESSES
                        Number of demultiplexing processes to spawn during execution.
  --n_writers N_WRITERS
                        Number of demultiplexing writing processes to spawn during execution.
  --no_gz               Do not compress (gzip) output FASTQ files.
  --undetermined_fq     Save Undetermined/Unselected/Short reads to Undetermined FASTQ files.
  --i7_rc {auto,true,false}
                        Reverse-complement I7 Index (Full Length protocol only). Enter "auto" to detect and auto-correct the
                        reverse complementation of I5/I7 indices by certain Illumina sequencers. Otherwise manually override
                        with "True" or "False").
  --i5_rc {auto,true,false}
                        See help section for "--i7_rc".
  --read_buffer READ_BUFFER
                        Buffer size of data sent to each demultiplexing (worker) process in GB.
  --prog PROG           Number of reads to process before updating status in log file.
  --no_split_fastqs     Output merged FASTQ file(s). Barcodes are written into read names and merged into large FASTQ file. By
                        default output into barcode-level FASTQ files.
  --use_barcodes USE_BARCODES
                        Limit number of barcodes to this value.
  --check_reads CHECK_READS
                        Use this number of reads to estimate read counts during barcode selection.
  --min_reads MIN_READS
                        Discard barcodes with estimated read count lower than this number.
  --random_pick         Pick random reads during barcode selection, rather than anazlyzing the first N read pairs.
  --preview             Print nextflow command without executing it.
```

Figure 10. The output of `cogent dna demux -h` at the command line.

The full list of dna analyze control options are listed in Table 7 and a screenshot of the output of $COGENT_AP_HOME/cogent dna analyze -h is shown in Figure 11

Table 7. Full list of options under `cogent dna analyze -h`.

Option	Description	Default
`-h, --help`	Produce a help message and exit the application	N/A
`-g, --genome`	Allows for selection of a supported genome or custom genome that you have installed	N/A
`-G, --genome_dir`	Specifies the directory where the genome and index files are installed.	`$COGENT_AP_HOME/genomes`
`-B {}, --bin_size{}`	Specifies the bin size used for Ginkgo analysis	N/A
`-r {}, --read_length {}`	Specifies the read length of the input data	N/A
`-R, --read_filter`	Specifies the minimum number of paired-end reads required per barcode to retain for downstream analysis	25,000
`-b, --barcodes_file`	Specifies path to the well-list file from CellSelect Software or another custom file containing only barcodes that were selected for sequencing	N/A
`-o, -output_dir`	Specifies the output directory in which to store the results of the pipeline	N/A
`-i, --input_dir`	Specifies the input directory that contains the results from 'rna demux'	N/A
`-t {}, --type_of_experiment {}`	Specifies experimental protocol	N/A
`--resume`	Resumes a previous pipeline run with the same inputs.	N/A
`--preview`	Prints Nextflow command without executing it.	N/A

```
usage: cogent dna analyze [-h] -g {hg38,mm39} [-G GENOME_DIR] -B {500kb,1mb} -r {76bp,151bp} [-R READ_FILTER]
                          -b BARCODES_FILE -o OUTPUT_DIR -i INPUT_DIR -t {shasta_wga} [--preview] [--resume]

            Script to perform CNV analysis by fastq input data.
            The input to this script are files output by Cogent demux.
            The modules currently included are:
                - Trimming (Trimmomatic)
                - Alignment (Bowtie2)
                - Sequencing QC (Picard/Samtools)
                - Summarization (TBUSA)
                - Reporting (TBUSA, CogentDS)

options:
  -h, --help            show this help message and exit
  -g {hg38,mm39}, --genome {hg38,mm39}
                        Select a supported genome or provide the name of a custom genome that you installed.
  -G GENOME_DIR, --genome_dir GENOME_DIR
                        Directory where genome and index files were installed by add_genome. [Default:
                        $COGENT_ROOT/genomes]
  -B {500kb,1mb}, --bin_size {500kb,1mb}
                        Bin size used for Ginkgo analysis.
  -r {76bp,151bp}, --read_length {76bp,151bp}
                        Read length of input data.
  -R READ_FILTER, --read_filter READ_FILTER
                        Minimum number of PE reads required per barcode to keep in downstream analysis.
                        [Default: 25000]
  -b BARCODES_FILE, --barcodes_file BARCODES_FILE
                        Well List file from Takara's CellSelect Software (Recommended), or another custom file
                        containing only barcodes that were selected for sequencing.
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Name of output directory to store results.
  -i INPUT_DIR, --input_dir INPUT_DIR
                        Directory contains results from demux command. The directory must contain FASTQ files
                        after demultiplexing.
  -t {shasta_wga}, --type_of_experiment {shasta_wga}
                        Experimental protocol used.
  --preview             Print nextflow command without executing it.
  --resume              Resume a previous pipeline run with the same inputs
```

Figure 11. The output of `cogent analyze -h` at the command line.

2. Adding a Genome Build

The human and mouse genome builds available from our server (Section IV.C, "Install Cogent NGS Analysis Pipeline v3.1) are recommended for use in the pipeline, but genomes of other species can be added into the software post-install.

To add custom genome data to CogentAP:

1. Create a copy of the file under

   `$COGENT_AP_HOME/config/genome_sources/sample.config`

   and rename it

   `$COGENT_AP_HOME/config/genome_sources/<common_species_name>.config`

   where `<common_species_name>` is the name of the genome being added (e.g., `dm6`)

2. Update the following fields using a text editor:

   - Replace 'GENOME' with the `<common_species_name>` from Step 1 (e.g., `dm6`)

- Replace 'ENSEMBL_GENOME_FASTA_URL' with the public URL of the FASTA file containing all the sequences (chromosomes and contigs) from Ensembl.

  Using the fruit fly genome from Ensembl.org as an example, you would replace 'ENSEMBL_GENOME_FASTA_URL' with the following URL:

  ```
  https://ftp.ensembl.org/pub/release-
  113/fasta/drosophila_melanogaster/dna/Drosophila_melanogaster.
  BDGP6.46.dna_sm.toplevel.fa.gz
  ```

- Replace 'PATH_TO_GINKGO_REF' with the path to the directory containing the Ginkgo reference files for the genome.

NOTES:

– Ginkgo reference files need to be generated for the newly added genome. For instructions on generating these files, please refer to https://github.com/robertaboukhalil/ginkgo/tree/master/genomes/scripts.

– For DNA-seq analysis, the lines 'annotation_gtf_url', 'sortmerna_fastas', and 'mito_reads' in the config file can be safely removed

3. Run the script:

```
$COGENT_AP_HOME/cogent dna add_genome \
  -g <common_species_name> \
  -G <genome_dir>
```

where `<common_species_name>` is the name of the genome being added, and `<genome_dir>` is the directory where the new genome needs to be stored in. If `-G` is not used, the genome is stored by default in `$COGENT_AP_HOME/genomes`.

For additional help with this script, type:

```
$COGENT_AP_HOME/cogent dna add_genome -h
```

CogentAP should now be able to analyze data with the genome option `-g` set to `<common_species_name>` during the `cogent dna analyze` step.

3. Processing Time

The time taken by the pipeline will vary based on the hardware specifications of the server on which it is run, the size of the raw-fastq input files, and where the files are stored. Expect analysis to take between 2–24 hr depending on the sequencing platform, the depth of the sequencing data, and the number of barcodes that are being analyzed.

If the raw-fastq files are instead stored on a network drive, these baselines might be exceeded.

D. Resuming an Analysis

CogentAP v3.1 is built on the popular Nextflow pipelining framework that allows for the option to resume a failed and/or stopped analysis. To resume an analysis that stopped in the middle of the run, the `--resume` flag can be added to the end of the same command used to run the analysis in the first place. If all the parameters are identical and the `work` directory that Nextflow created is left intact, the analysis should commence from the step where it stopped.

E. Clearing Out the Work Directory

It is generally a good idea to delete the Nextflow work directory on a regular basis. Even though CogentAP deletes all intermediate files upon successful completion of an analysis, data from older failed or incomplete analyses are sometimes preserved in the working directory, taking up unnecessary space. In such cases, cleaning out the work directory could help reclaim space.

VII. Test Dataset

A mini dataset file (referred to here as test dataset) is included in the CogentAP distribution package; it can be found under the main installation folder in the following sub-folder (Figure 12, below):

`$COGENT_AP_HOME/test/fixtures/experiments/ICELL8_FLA.`

This dataset can be used to test the running of the pipeline end-to-end and will provide a sample of the output files. The output (report and stats only) from the test dataset is also included in the CogentAP installation and can be found in the following folder:

`$COGENT_AP_HOME/test/fixtures/experiments/ICELL8_FLA/out_test/.`

These output files can be used to compare to the output of your test run to verify everything is working correctly.

NOTE: The test dataset should not be used for inference purposes. CogentAP output statistics and plots will only be meaningful with a real dataset.

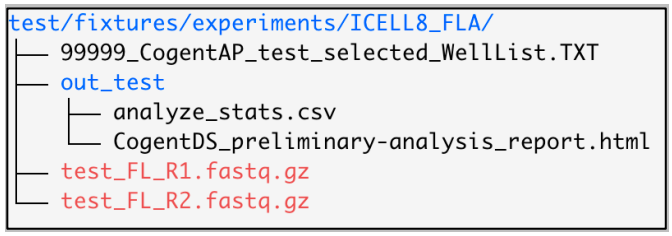


Figure 12. The `test/fixtures/experiments/ICELL8_FLA` folder under `$COGENT_AP_HOME`. The sample `*.fastq.gz` files and example output directory `out_test/` can be found there.

To start a run using test data, use the following commands to run the RNA demux and RNA analyze, respectively:

```
cd $COGENT_AP_HOME
```

```
cogent rna demux \
 -f test/fixtures/experiments/ICELL8_FLA/test_FL_R1.fastq.gz \
 -p test/fixtures/experiments/ICELL8_FLA/test_FL_R2.fastq.gz \
 -b test/fixtures/experiments/ICELL8_FLA/99999_CogentAP_test_selected_WellList.TXT \
 -t icell8_fla -o out_test
```

```
cogent rna analyze \
  -i out_test/demultiplexed_fastqs/ -t icell8_fla \
  -o out_test/analyze -g hg38
```

The test run should take ~5–10 min to complete successfully.

VIII. Output Files

The pipeline produces output files that serve two purposes:

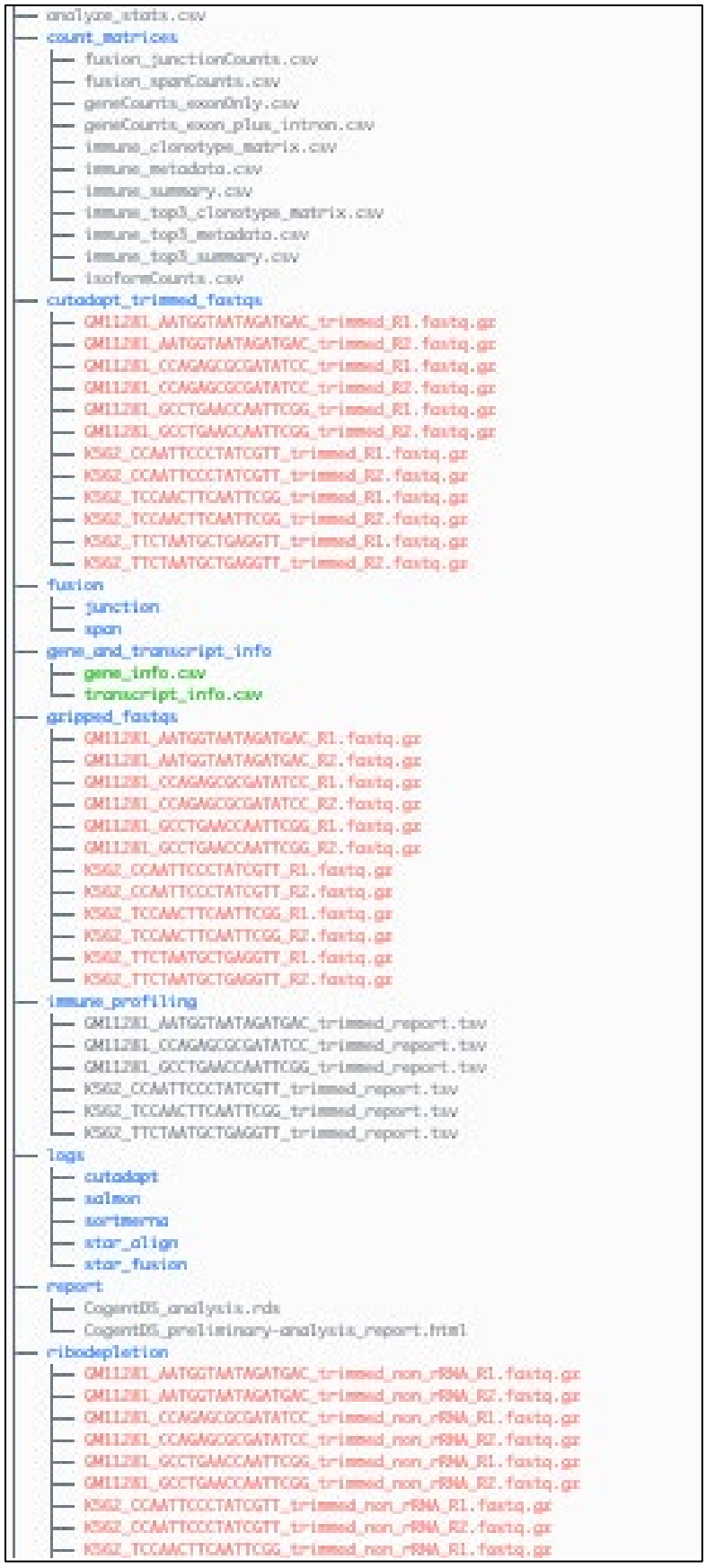
1. Summarization of results using typical statistics and plots
2. Facilitating further analyses using our interactive R kit, Cogent NGS Discovery Software (CogentDS), or any other tertiary analysis tool

A. Output Folder Structure

The folder structure of the results folder is slightly different depending on which analysis was run and which kit was used to generate sequencing data.

1. RNA-Seq Analysis Output (for All RNA-Seq Kits Except the Shasta Total RNA-Seq Kit)

For RNA-seq analysis of sequencing data produced from all RNA-seq kits except the Shasta Total RNA-Seq Kit, the contents of the demux output folder and analysis output folders will resemble Figure 13 (below).


analyze_stats.csv
count_matrices
fusion_junctionCounts.csv
fusion_spanCounts.csv
geneCounts_exonOnly.csv
geneCounts_exon_plus_intron.csv
immune_clonotype_matrix.csv
immune_metadata.csv
immune_summary.csv
immune_top3_clonotype_matrix.csv
immune_top3_metadata.csv
immune_top3_summary.csv
isoformCounts.csv
cutadapt_trimmed_fastqs
GM11281_AATGGTAATAGATGAC_trimmed_R1.fastq.gz
GM11281_AATGGTAATAGATGAC_trimmed_R2.fastq.gz
GM11281_CCAGAGCGCGATATCC_trimmed_R1.fastq.gz
GM11281_CCAGAGCGCGATATCC_trimmed_R2.fastq.gz
GM11281_GCCTGAACCAATTCGG_trimmed_R1.fastq.gz
GM11281_GCCTGAACCAATTCGG_trimmed_R2.fastq.gz
K562_CCAATTCCCTATCGTT_trimmed_R1.fastq.gz
K562_CCAATTCCCTATCGTT_trimmed_R2.fastq.gz
K562_TCCAACTTCAATTCGG_trimmed_R1.fastq.gz
K562_TCCAACTTCAATTCGG_trimmed_R2.fastq.gz
K562_TTCTAATGCTGAGGTT_trimmed_R1.fastq.gz
K562_TTCTAATGCTGAGGTT_trimmed_R2.fastq.gz
fusion
junction
span
gene_and_transcript_info
gene_info.csv
transcript_info.csv
gzipped_fastqs
GM11281_AATGGTAATAGATGAC_R1.fastq.gz
GM11281_AATGGTAATAGATGAC_R2.fastq.gz
GM11281_CCAGAGCGCGATATCC_R1.fastq.gz
GM11281_CCAGAGCGCGATATCC_R2.fastq.gz
GM11281_GCCTGAACCAATTCGG_R1.fastq.gz
GM11281_GCCTGAACCAATTCGG_R2.fastq.gz
K562_CCAATTCCCTATCGTT_R1.fastq.gz
K562_CCAATTCCCTATCGTT_R2.fastq.gz
K562_TCCAACTTCAATTCGG_R1.fastq.gz
K562_TCCAACTTCAATTCGG_R2.fastq.gz
K562_TTCTAATGCTGAGGTT_R1.fastq.gz
K562_TTCTAATGCTGAGGTT_R2.fastq.gz
immune_profiling
GM11281_AATGGTAATAGATGAC_trimmed_report.tsv
GM11281_CCAGAGCGCGATATCC_trimmed_report.tsv
GM11281_GCCTGAACCAATTCGG_trimmed_report.tsv
K562_CCAATTCCCTATCGTT_trimmed_report.tsv
K562_TCCAACTTCAATTCGG_trimmed_report.tsv
K562_TTCTAATGCTGAGGTT_trimmed_report.tsv
logs
cutadapt
salmon
sortmerna
star_align
star_fusion
report
CogentDS_analysis.rds
CogentDS_preliminary-analysis_report.html
ribodepletion
GM11281_AATGGTAATAGATGAC_trimmed_non_rRNA_R1.fastq.gz
GM11281_AATGGTAATAGATGAC_trimmed_non_rRNA_R2.fastq.gz
GM11281_CCAGAGCGCGATATCC_trimmed_non_rRNA_R1.fastq.gz
GM11281_CCAGAGCGCGATATCC_trimmed_non_rRNA_R2.fastq.gz
GM11281_GCCTGAACCAATTCGG_trimmed_non_rRNA_R1.fastq.gz
GM11281_GCCTGAACCAATTCGG_trimmed_non_rRNA_R2.fastq.gz
K562_CCAATTCCCTATCGTT_trimmed_non_rRNA_R1.fastq.gz
K562_CCAATTCCCTATCGTT_trimmed_non_rRNA_R2.fastq.gz
K562_TCCAACTTCAATTCGG_trimmed_non_rRNA_R1.fastq.gz

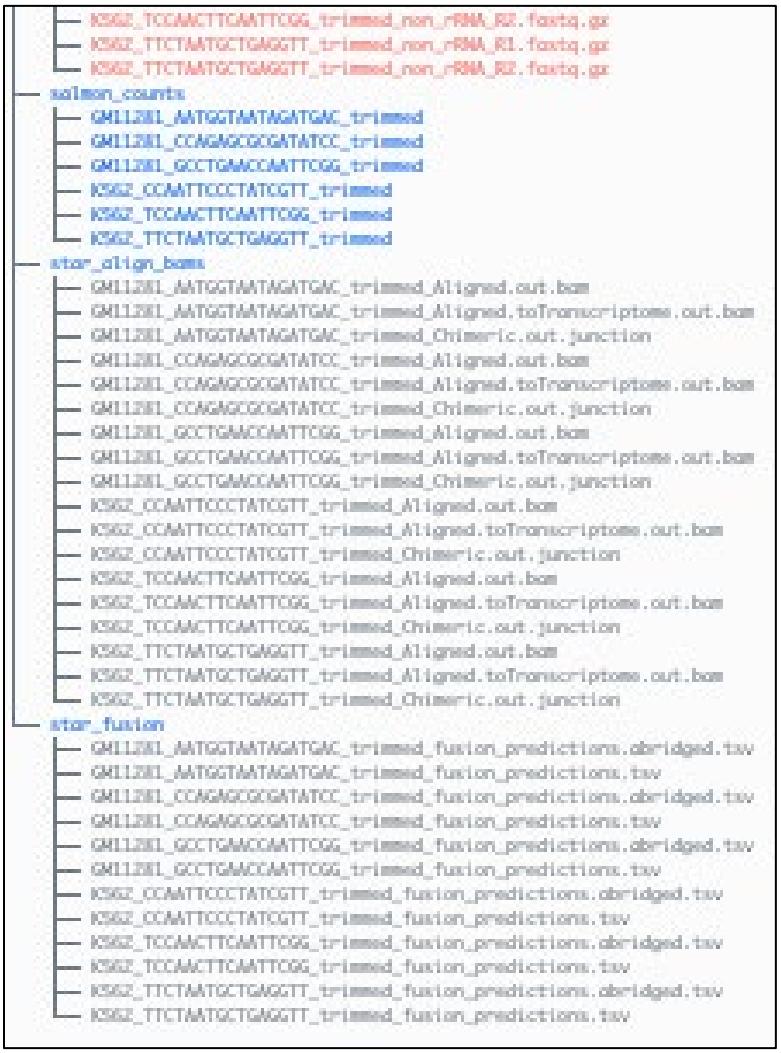


Figure 13. Folders and files of the output directory for a typical RNA-seq analysis.

2. RNA-Seq Analysis Output (for the Shasta Total RNA-Seq Kit)

For RNA-seq analysis of Shasta Total RNA-Seq Kit data, the demux output folder remains the same as shown in Figure 13. However, due to the sheer volume of files that get created and the amount of storage they take, only the analyze stats file, the count matrices, the report, and an Rda object created for use with CogentDS are saved in the output folder by default, as shown in Figure 14.

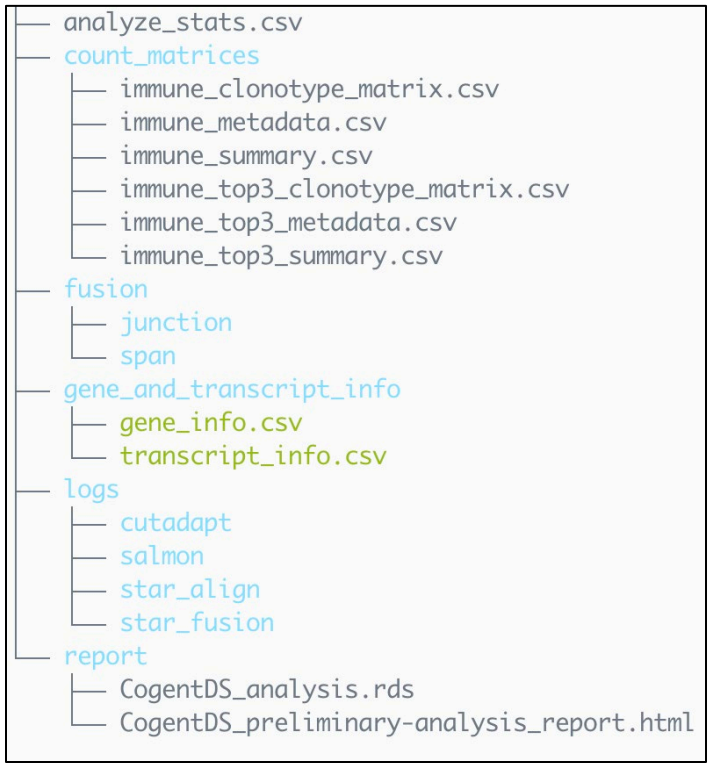


Figure 14. Folders and files of the output directory for a Shasta Total RNA-Seq Kit analysis. This example also includes the optional immune and fusion analyses (using the default parameters).

If all the files from the analysis need to be saved, the `--keep_intermediate` flag can be used. This will add considerable overhead, both in analysis time, and in storage space utilized. Therefore, it is not recommended unless necessary. The directory structure from such an analysis will resemble the folder shown in the previous section in Figure 13.

3. DNA-Seq Analysis Output

For a typical DNA-seq analysis, the contents of the demux output folder and analysis output folders will resemble Figures 15 and 16:

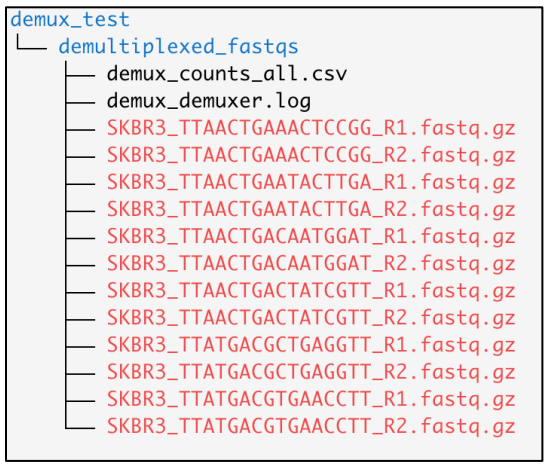


Figure 15. Folders and files of the directory for a typical Cogent DNA demux output folder.

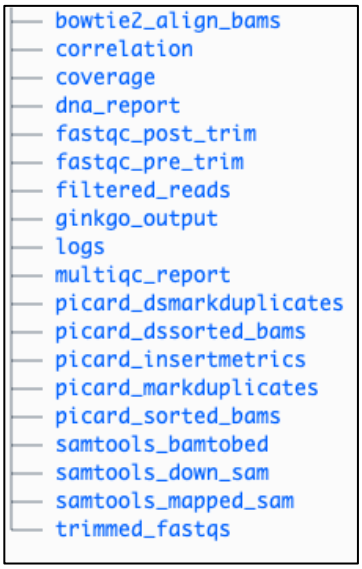


Figure 16. Folders of the directory for a typical Cogent DNA-seq analysis output folder.

B. HTML Report

HTML reports are generated by the same report process as CogentDS, using standard parameters, and contain the example statistics and plots listed below. For complete details, please see the Cogent NGS Discovery Software User Manual.

1. RNA-seq Analysis

NOTE: Some sections and/or plots may not be generated depending on the sample size and quality of data provided to CogentAP. Ribosomal read counts could either be present or absent in the report based on the kit used and if in silico ribodepletion was enabled during the analysis.

a) Experimental Overview and Data Statistics Plot

Experimental Overview

Read Stats

	Total Counts	% (of Barcoded Reads)	% (of Trimmed Reads)
Barcoded_Reads	446,464,559	100.00	NA
Trimmed_Reads	423,817,645	94.93	100
Unmapped_Reads	18,443,687	4.13	4.35
Mapped_Reads	405,373,958	90.80	95.65
Uniquely_Mapped_Reads	390,581,665	87.48	92.16
Multimapped_Reads	14,792,293	3.31	3.49
Chimeric_Reads	28,115,071	6.30	6.63
Mitochondrial_Reads	15,478,999	3.47	3.65
Usable	389,894,959	87.33	92
Undesirable	15,478,999	3.47	3.65

Gene Body Assignment Breakdown

	Total Counts	% (of Mapped Reads)
Mapped_Reads	405,373,958	100.00
Exon_Reads	109,238,159	26.95
Intron_Reads	252,050,458	62.18
Gene_Reads	361,288,614	89.12
Intergenic_Reads	44,085,344	10.88

Undesirable Read Breakdown

	Total Counts	% (of Mapped Reads)
Mapped_Reads	405,373,958	100.00
Mitochondrial_Reads	15,478,999	3.82
Usable	389,894,959	96.18

Other Stats

	Average Stats across barcodes
No_of_Genes	2,420
No_of_Transcripts	1,205
Strand_Specificity	1

Figure 17. Example experimental overview section of the HTML report.

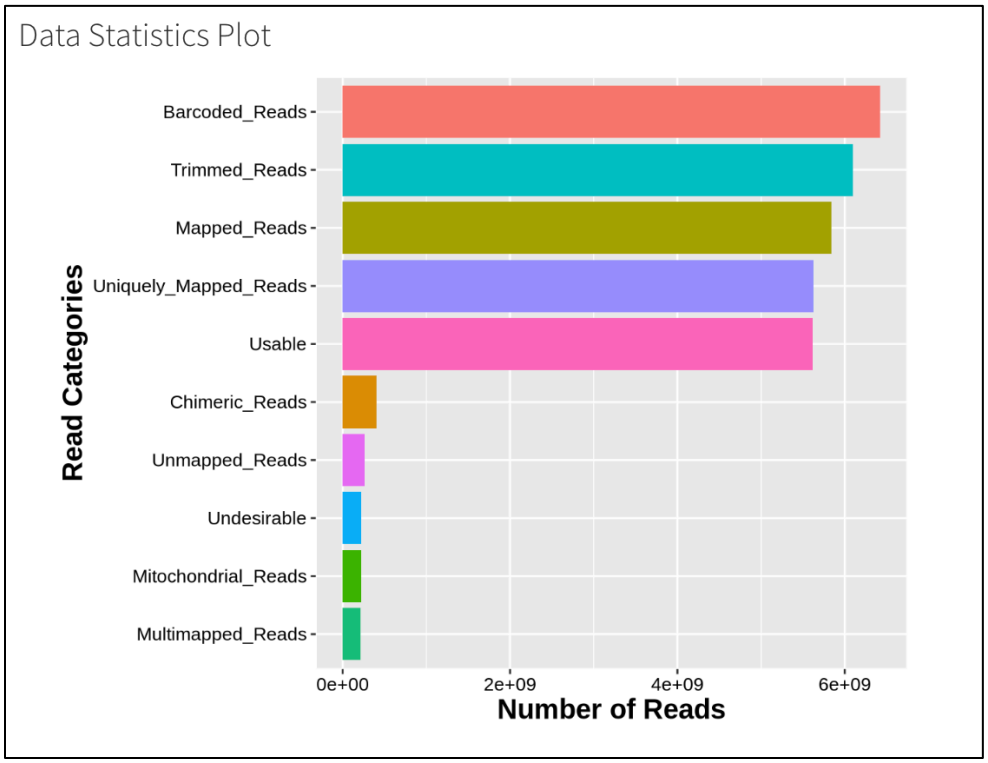


Figure 18. Example data statistics plot from the HTML report.

b) QC Analysis

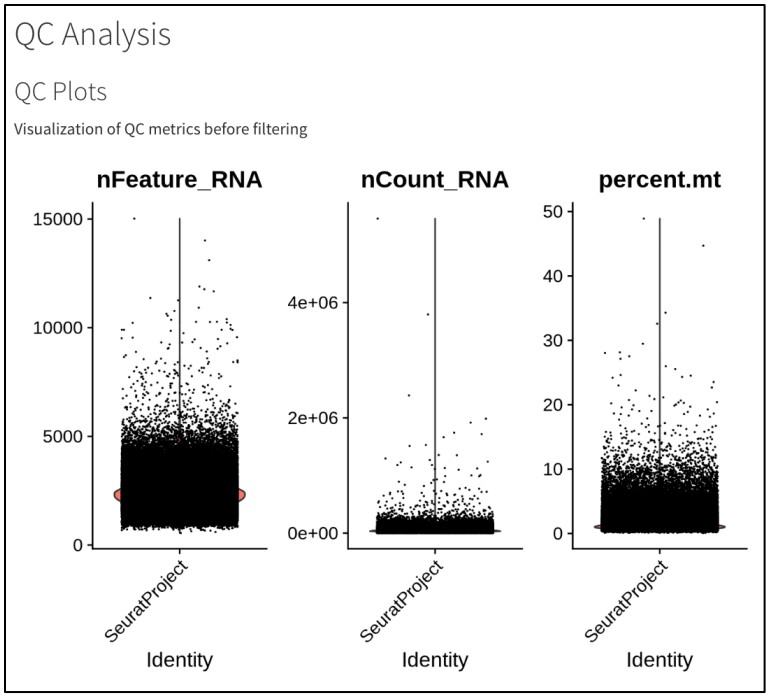


Figure 19. Example QC analysis section of the HTML report.

c) Principal Component Analysis (PCA)

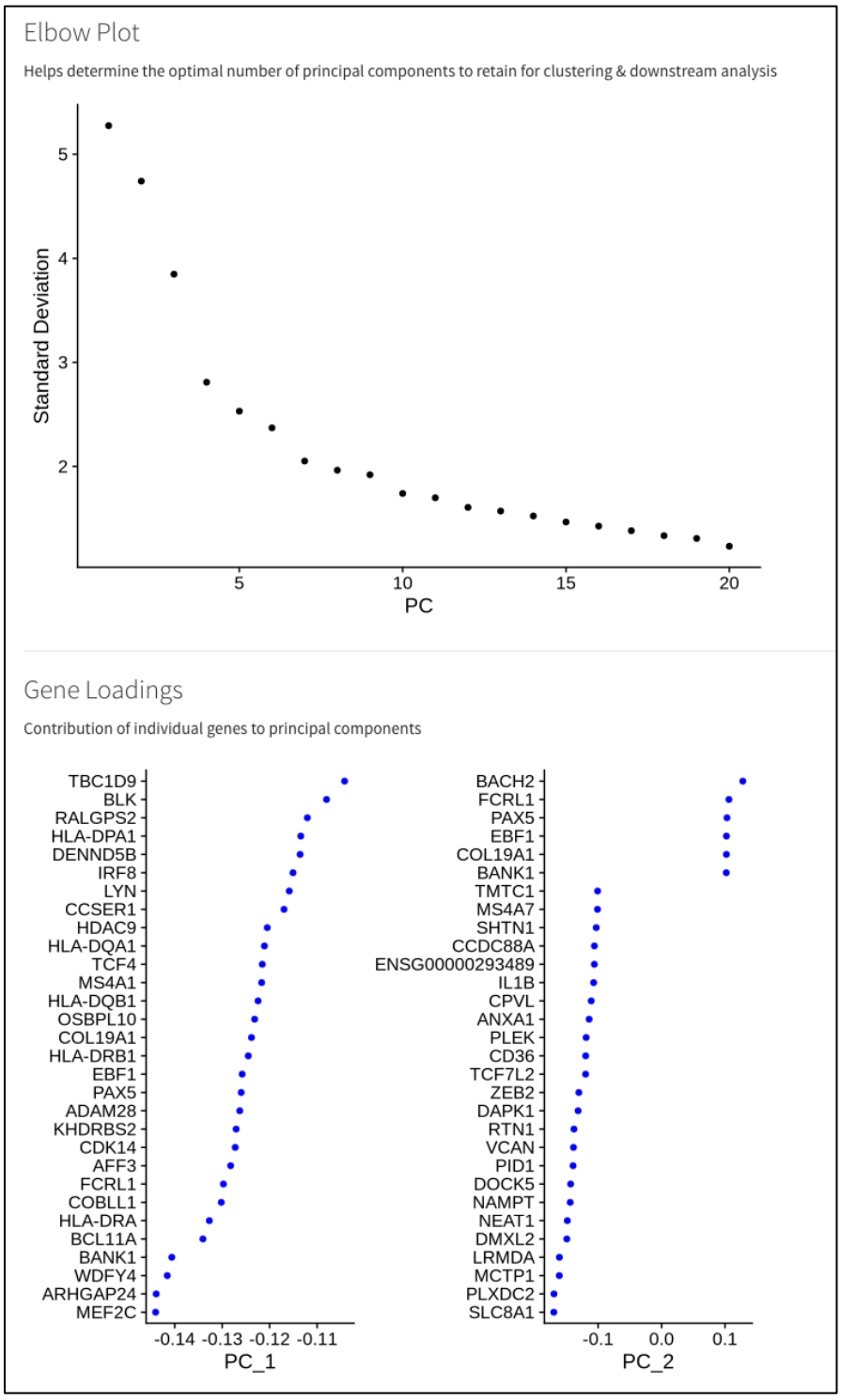


Figure 20. Example PCA analysis plots from the HTML report.

d) UMAP Plot

NOTE: UMAP plots are only generated for sequencing data from single-cell RNA-seq kits.

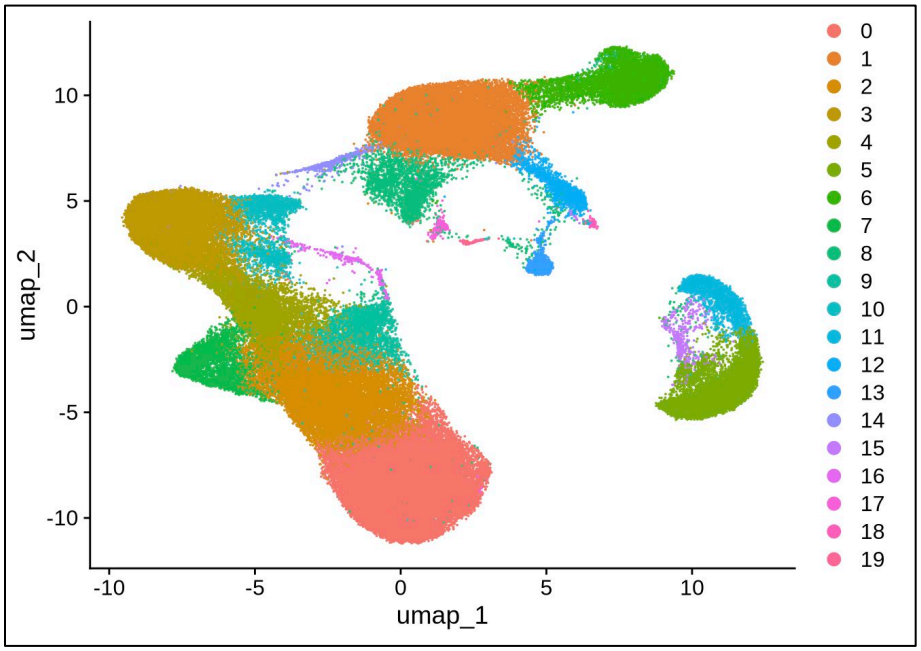


Figure 21. Example UMAP plot from the HTML report.

2. DNA-Seq Analysis

NOTE: Some sections and/or plots may not be generated depending on the sample size and quality of data provided to CogentAP.

Analysis of sequencing data from DNA-seq or Shasta WGA kits results in the generation of two reports—a QC metrics report and an analysis report.

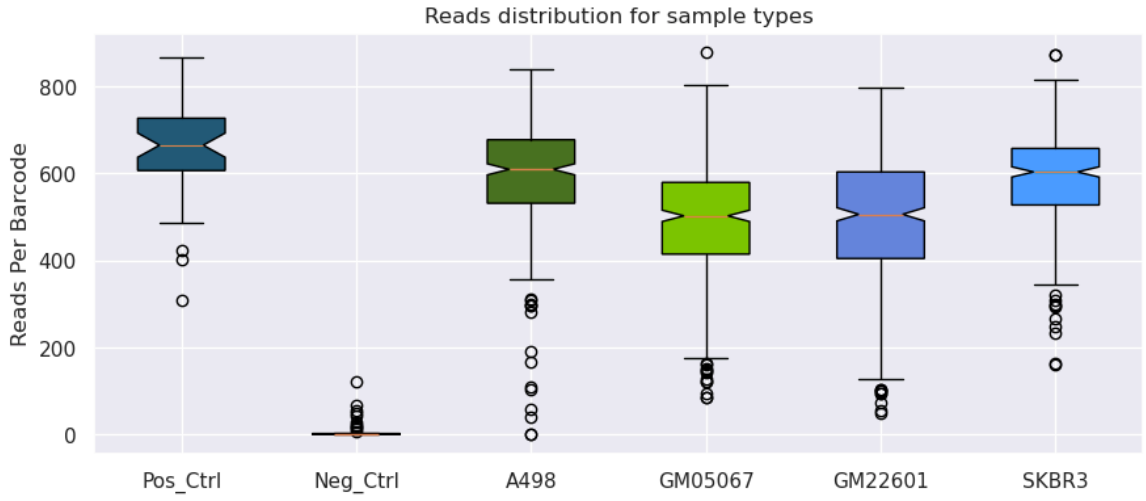
a) QC Metrics Report

Experimental Overview and Reads by Sample Type

Experimental Overview

Total Read pairs	1,000,000
Total Reads	2,000,000
Total Barcoded Reads	1,584,150
Barcoded Reads Fraction	79.21%
Total Barcodes Identified	1,527
Min Read Pairs Filter	500

Reads by Sample Type



Sample Types	Pos_Ctrl	Neg_Ctrl	A498	GM05067	GM22601	SKBR3
Barcodes Identified	46	46	321	393	424	297
Barcodes Reads	60,154	866	379,296	380,432	416,068	347,334
% of Total Barcoded Reads	3.8%	0.05%	23.94%	24.01%	26.26%	21.93%
Reads per Barcode_median	1,329	0	1,218	1,004	1,011	1,206
Reads per Barcode_mean	1,308	19	1,182	968	981	1,169
Reads per Barcode_stdDev	220	45	266	266	297	231

Figure 22. Example Experimental Overview table and Reads by Sample Type plot from the QC metrics report.

Read Statistics and Additional Metrics

Reads Statistics wells after Min Reads Filter

Sample Types	Pos_Ctrl		A498		GM05067		GM22601		SKBR3	
Passing-filter Barcodes	42		266		200		220		240	
	Reads	% Barcoded Reads	Reads	% Barcoded Reads	Reads	% Barcoded Reads	Reads	% Barcoded Reads	Reads	% Barcoded Reads
Barcoded Reads	56,920	100.00%	338,122	100.00%	234,138	100.00%	265,764	100.00%	301,340	100.00%
Trimmed Reads	56,918	100.0%	338,094	99.99%	234,116	99.99%	265,742	99.99%	301,314	99.99%
Unmapped Reads	3,186	5.6%	17,142	5.07%	12,419	5.3%	14,428	5.43%	16,178	5.37%
Mapped Reads	53,732	94.4%	320,952.0	94.92%	221,697.0	94.69%	251,314.0	94.56%	285,136.0	94.62%
Duplicate Reads	301	0.53%	1,635.0	0.48%	1,080.0	0.46%	1,189.0	0.45%	1,452.0	0.48%
Unique Reads	53,431	93.87%	319,317.0	94.44%	220,617.0	94.23%	250,125.0	94.12%	283,684.0	94.14%

Additional Metrics

Sample Types	Pos_Ctrl	A498	GM05067	GM22601	SKBR3
Insert Size (mean)	261.94	261.96	260.10	260.32	261.78
Insert Size (median)	236.50	237.00	236.00	237.00	236.00
GC Content (mean)	40.83%	40.16%	40.41%	40.44%	40.25%
GC Content (median)	41.00%	40.00%	40.00%	40.00%	40.00%
Normalized Reads per Chromosome (mean)	1.00	0.97	1.00	1.00	0.99
Normalized Reads per Chromosome (stdev)	0.10	0.27	0.13	0.12	0.43
Pearson's Correlation (mean)	0.02	0.04	0.02	0.02	0.14
Pearson's Correlation (stdev)	0.02	0.02	0.02	0.02	0.04

Figure 23. Example Read Statistics and Additional Metrics tables from the QC metrics report.

b) Analysis Report

QC Analysis Plots

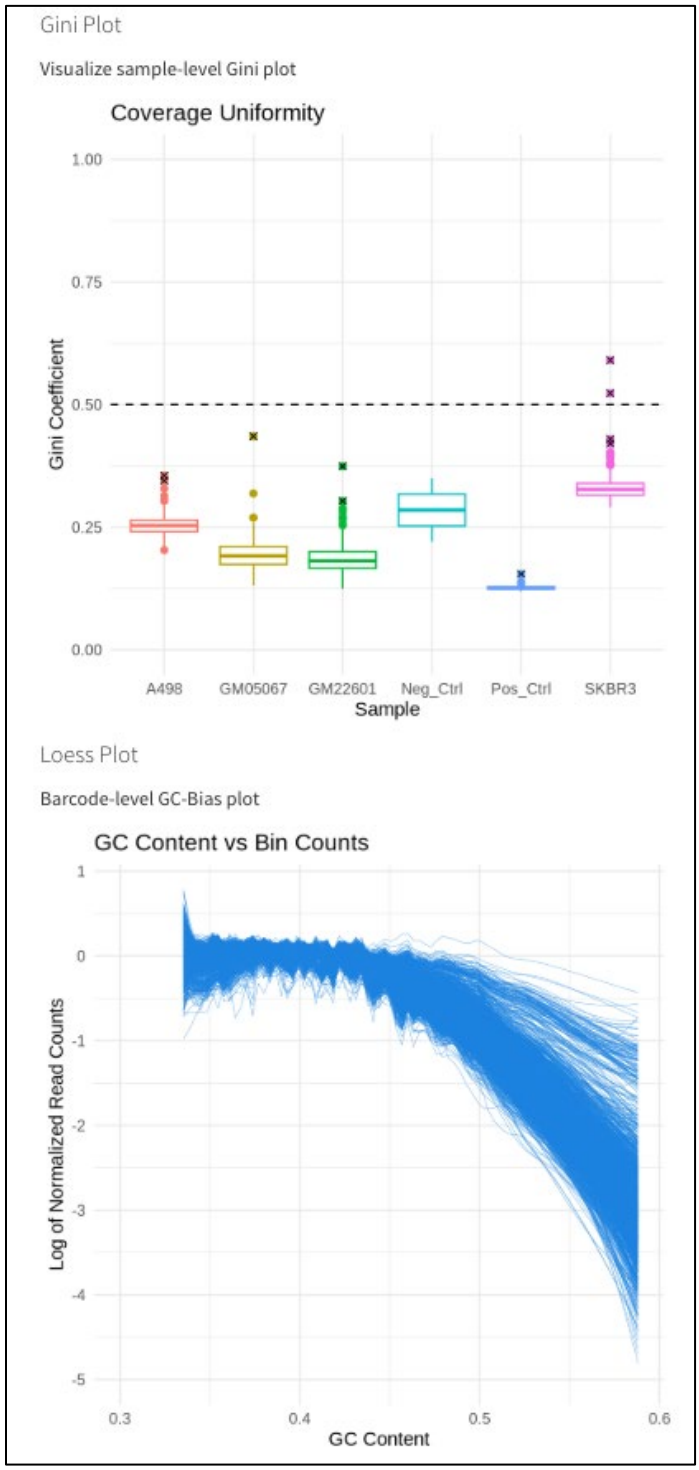


Figure 24. Example Gini Plot and Loess Plot from the DNA-seq analysis report.

CCN Heatmap

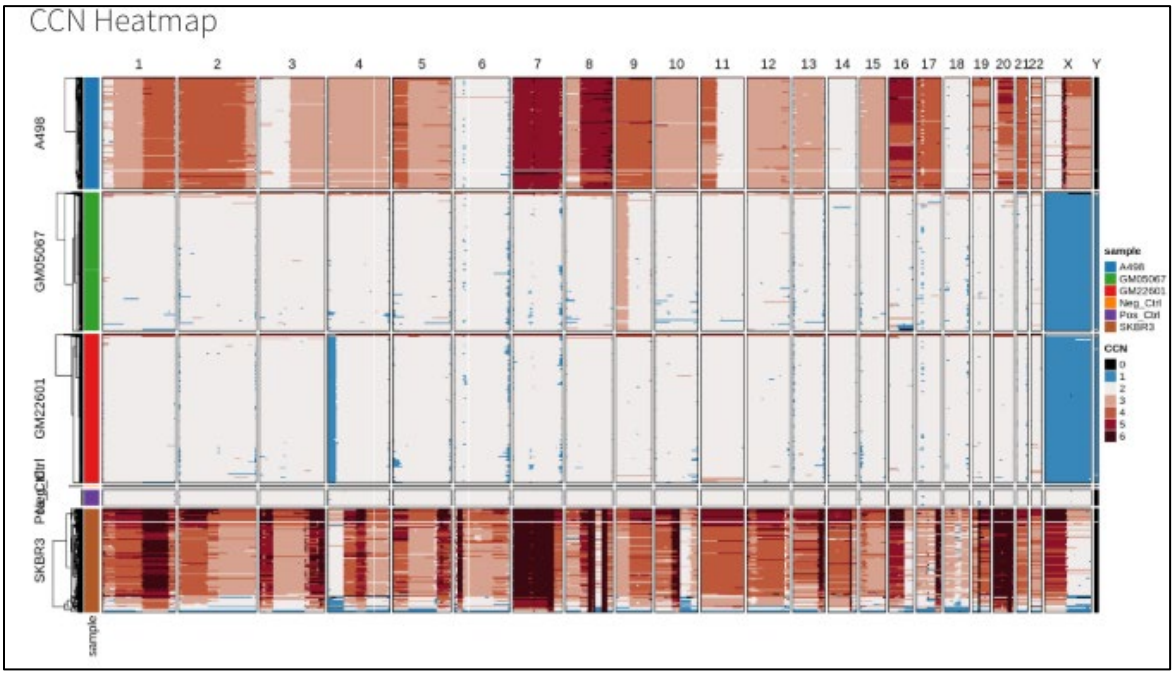


Figure 25. Example CCN Heatmap plot from the DNA-seq analysis report.

UMAP Plot

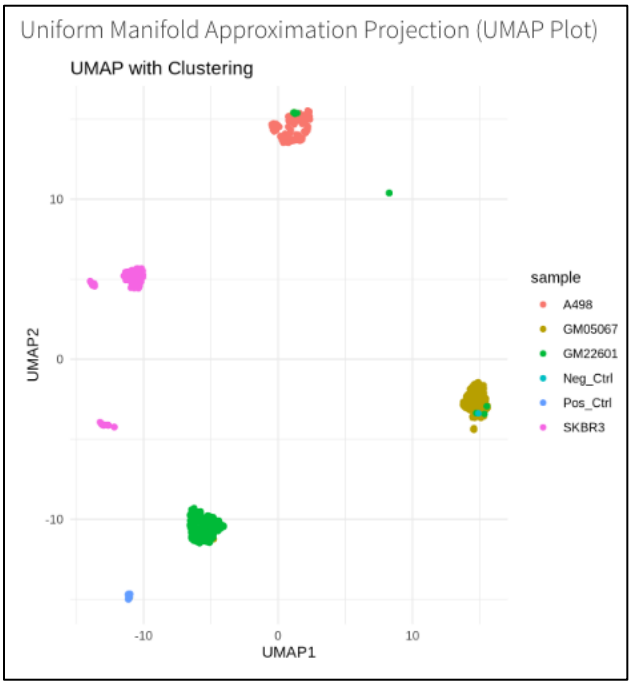


Figure 26. Example UMAP plot from the DNA-seq analysis report.

3. CogentDS Analysis Rdata Object Files

During the generation of the HTML report, an Rdata file is created with the results of the various analysis modules. This file can be used directly as input into CogentDS to perform additional analysis, saving processing time in that tool. The name of the file depends on which analysis is

being run; for RNA-seq analysis, the file is called `CogentDS_analysis.rds`; for DNA-seq analysis, it is called `CogentDS_scDNA_analysis.rds`. The RNA-seq analysis rds file contains three assay objects: 'RNA' for exon and intron counts, 'Exon_RNA' for exon-only counts, and 'Transcript_RNA' for transcript counts.

C. Raw Data Files

1. RNA-Seq Analysis

CogentAP RNA-seq analysis generates several raw data files, based on the experiment type. The table below lists the possible raw output files grouped by analysis option as found within the output folder specified during the analysis run (Section VII.A). For more details about the files themselves, please refer to the Appendix.

Table 8. Raw data files generated by CogentAP RNA-seq analysis.

Analysis option	Referred to as	File name	Subfolder
(Default)	Overall Stats*	`analysis_stats.csv`†	
	Gene Info File	`gene_info.csv`	gene_and_transcript_info/
	Transcript Info File	`transcript_info.csv`	gene_and_transcript_info/
(Default)	Gene matrix	`geneCounts_exonOnly.csv`†	count_matrices/
	Gene matrix with intron counts	`geneCounts_exon_plus_intron.csv`†	
	Transcript matrix	`isoformCounts.csv`†	
Gene fusion analysis	Junction matrix	`fusion_junctionCounts.csv`†	count_matrices/
	Spanning matrix	`fusion_spanCounts.csv`†	count_matrices/
	Junction data	`junction/barcodes.tsv.gz`, `junction/features.tsv.gz`, `junction/matrix.mtx.gz`	fusion/
	Spanning data	`span/barcodes.tsv.gz`, `span/features.tsv.gz`, `span/matrix.mtx.gz`	fusion/
Immune profiling analysis	Clonotype matrix	`immune_clonotype_matrix.csv`	count_matrices/
	Metadata	`immune_metadata.csv`	
	Summary	`immune_summary.csv`	
	Top 3 clonotype matrix	`immune_top3_clonotype_matrix.csv`	
	Top 3 metadata	`immune_top3_metadata.csv`	
	Top 3 summary	`immune_top3_summary.csv`	

*When running data from plate-based full-length transcriptome analysis with UMIs (`smartseq_fla_umi`), two stats files may be generated. Please see the Appendix, Section A for more details.

†Only generated when the number of barcodes in the analysis is ≤5,000.

2. DNA-Seq Analysis

The following table lists the possible raw output files generated by CogentAP DNA-seq analysis as found within the output folder specified in the analysis run. For more information about the files themselves, please refer to the Appendix.

Table 9. Raw data files generated by CogentAP DNA-seq analysis.

Analysis option	Referred to as	File name	Subfolder
(Default)	Ginkgo Data	data	ginkgo_output/
	Ginkgo SegCopy	SegCopy	
	Ginkgo SegFixed	SegFixed	
	Ginkgo SegNorm	SegNorm	
	Ginkgo SegStats	SegStats	
(Default)	Multiqc General Stats	multiqc_general_stats.txt	multiqc_report/multiqc_report_data/
	Multiqc FastQC results	multiqc_fastqc.txt	
	Multiqc Alignment Stats	multiqc_bowtie2.txt	
	Multiqc Trimming Stats	multiqc_trimmomatic.txt	
	Multiqc Picard MarkDuplicates Stats	multiqc_picard_dups.txt	
	Multiqc Picard InsertSize Stats	multiqc_picard_insertSize.txt	

D. logs Folder

The logs/ folder contains the log files generated by various tools used in the pipeline. They can be used for debugging or reference purposes.

E. BAM Files

1. RNA-Seq Analysis

The star_align_bams/ folder contains BAM files generated during the alignment step in gene and transcript expression analysis. These files are required for gene expression, transcript expression, and gene fusion analysis.

- *.Aligned.out.bam—output files from genome alignment.
- *.Aligned.toTranscriptome.out.bam—output files containing transcriptome-based alignment information that are used for gene and transcript expression analysis.
- *.Chimeric.out.junction—output files with chimeric junction information used in gene fusion analysis

2. DNA-Seq Analysis

The bowtie2_align_bams/ folder contains BAM files generated during the alignment step in the WGA workflow.

Appendix A. Analysis of Raw RNA-Seq Data Files

NOTE: The information in this appendix only applies to RNA-seq analysis of sequencing data produced with RNA-seq kits, excluding the Shasta Total RNA-Seq Kit, and for analyses with ≤5,000 barcodes. The raw data output files listed in this appendix are all in CSV format.

A. Default Analysis Files

Table 10. Processed data output files generated by the default CogentAP analysis command for RNA-seq analysis.

Referred to as	File name
Stats	`analysis_stats.csv`
Gene matrix	`geneCounts_exonOnly.csv`
Gene matrix with intron counts	`geneCounts_exon_plus_intron.csv`
Gene info	`gene_info.csv`
Transcript matrix	`isoformCounts.csv`
Transcript info	`transcript_info.csv`

1. Stats File

The stats file contains barcode-level statistics across the analysis pipeline. Starting from barcoded reads, it summarizes the number of reads after each step in the pipeline: trimmed reads, mapped reads, exon/intron/intergenic reads, mitochondrial reads, ribosomal reads, etc. It also lists the number of genes detected per barcode.

Table 11 and Table 12 Below document all potential columns that might appear in the Stats file. Not all stats files will include every column listed, as the columns shown in this file depend on the reagent kit used to generate the input data.

Table 11. Columns that will be present in the `*_stats.csv` file output by CogentAP (input workflow agnostic).

Column name	Description
Barcode	Detected barcodes. This value will usually be the sample name from the well-list or well-list-like file, but there are three exceptions, documented in the table below.
Sample	Sample names described in sample description file. This column is used for grouping stats/plots in CogentDS and should be filled with the type of sample it is.
Barcoded_Reads	Number of reads after demultiplexing.
Trimmed_Reads	Number of remained reads after trimming.
Unmapped_Reads	Number of reads not mapped to genome.
Mapped_Reads	Number of reads mapped to genome.
Uniquely_Mapped_Reads	Number of reads mapped to one genomic location. These reads are used for counting.
Multimapped_Reads	Number of reads mapped to multiple genomic locations.
Chimeric_Reads	Number of reads that have multiple subsections align to multiple distinct portions of the genome with little or no overlap.
Exon_Reads	Number of reads assigned to an exonic region.
Intron_Reads	Number of reads assigned to an intronic region.

Column name	Description
Gene_Reads	Number of reads assigned to a gene region (exon + intron).
Intergenic_Reads	Number of reads assigned to an intergenic region.
No_of_Genes	Number of detected genes.
No_of_Transcripts	Number of detected transcripts.
Ribosomal_Reads	Number of reads assigned to a ribosomal gene.
Mitochondrial_Reads	Number of reads assigned to mitochondrial chromosome.

Table 12 lists additional columns that will be present in the stats file when the input FASTQ files result from any kits with UMIs. Two types of output files will be generated: one using 5′ UMI-reads and one using all reads (both 5′ UMI and internal non-UMI reads, all combined for non-UMI analysis). The columns below are shown only in the stats file for the 5′ reads.

Table 12. Additional columns in the stats file protocols that utilize UMIs in the workflow.

Column name	Description
No_of_Reads_w_UMI	Number of reads containing UMIs. It means the reads are 5″ reads.
Reads_After_Dedup_nUMIs_USSs	Number of reads left after deduplication with UMI tools.
Exon_nUMIs_USSs	Number of deduplicated reads assigned to an exonic region. Deduplication is done by both UMI and USS.
Intron_nUMIs_USSs	Number of deduplicated reads assigned to an intronic region. Deduplication is done by both UMI and USS.
Gene_nUMIs_USSs	Number of deduplicated reads assigned to a gene region (exon + intron). Deduplication is done by both UMI and USS.
Strand_Specificity	Ratio of reads detected as correct strand after mapping to genome.

2. Gene Matrix File

The gene matrix file (also referred to as the gene table or counts matrix) is also in CSV format and contains gene counts for each barcode, with the genes in the rows and barcodes in the columns. The file contains raw counts that can then be normalized and transformed using CogentDS. An example is shown below.

	AACCGGTTAATATTCG	AACCGGTTACTTCTAC	AACCGGTTAGAAGTAA	AACCGGTTAGCATTGA	AACCGGTTAGGCCAAG	AACCGGTTCAATGGAT
5S-rRNA	0	0	0	0	0	0
5-8S-rRNA	0	0	0	0	0	0
7SK	0	0	0	0	0	0
A1BG	0	5	0	0	0	0
A1BG-AS1	0	0	0	0	0	0
A1CF	0	0	0	0	1	0
A2M	0	0	0	5	328	3
A2M-AS1	0	0	0	0	0	0
A2ML1	0	0	0	1	0	0
A2ML1-AS1	0	0	0	0	0	0
A2ML1-AS2	0	0	0	0	0	0
A2MP1	0	0	0	0	17	0
A3GALT2	0	0	0	0	0	0
A4GALT	0	0	0	0	0	1
A4GNT	0	0	0	0	0	0
AA06	0	0	0	0	0	0
AAAS	0	1	0	0	0	0
AACS	1	0	0	0	0	0
AACSP1	0	0	0	1	0	0
AADAC	0	0	0	0	0	0
AADACL2	0	0	0	0	0	0
AADACL2-AS1	3	0	0	0	1	0
AADACL3	0	0	0	0	0	0
AADACL4	0	0	0	0	0	0
AADACP1	0	0	0	0	0	0
AADAT	0	0	0	0	0	0
AAGAB	1	0	0	0	77	10
AAK1	2	6	21	54	9	34
AAMDC	0	0	0	0	0	4
AAMP	0	32	0	0	12	15

Figure 27. Example of a gene matrix file.

3. Gene Matrix File Including Intron Counts

The gene matrix file including intron counts contains gene counts for each barcode with intron counts added to them, with the genes in the rows and barcodes in the columns. An example is shown below.

	AAGAGCGCAACTTAAC	AAGAGCGCACCGAATT	AAGAGCGCAGAAGTAA	AAGAGCGCAGACCGTT	AAGAGCGCAGATTCAT	AAGAGCGCCAATCTTG	AAGAGCGCCAATGGAT
ABCA15P	0	3	0	2	4	0	2
ABCA17P	1	10	0	0	5	1	5.001
ABCA2	0	0	0	0	0	1	0
ABCA3	3	2	1	5	2	9	1
ABCA3P1	0	0	0	0	0	0	0
ABCA4	2	9	1	8	3	1	7
ABCA5	3	17	0	9	15.53	6	16
ABCA6	15	9	2	2	13	1	3
ABCA7	14	0	0	1	0	0	0
ABCA8	1	10.504	0	1	9	4	5.025
ABCA9	2	8.187	3	2	3.145	7.638	5.769
ABCA9-AS1	0	13.309	0	0	6.855	1.362	10.207
ABCB1	25.266	31.101	0	7.194	27.25	7.663	12.259
ABCB10	1	3	13	1	3	2	3
ABCB10P1	0	0	3	0	0	0	0
ABCB10P3	0	0	0	0	0	0	0
ABCB10P4	0	0	0	0	0	0	0
ABCB11	5	12	3	7	27	7	10
ABCB4	11	8	2	4	8	6	6
ABCB5	6	23	2	20	14	5	14
ABCB6	5	0	0	0	0	0	0
ABCB7	1	6	1	0	6	4	6
ABCB8	1	0	0	0	1	0	0
ABCB9	0	3	1	0	5.04	0	3
ABCC1	17	10	5	3	6	6	16
ABCC10	2	1.012	0	0	0	0	0
ABCC11	0	5	1.014	0	1	0	3
ABCC12	0	3	0	0	2	0	3
ABCC13	6	16.001	3	9	13	8	20
ABCC2	2.002	14	2	5	5	2.004	4
ABCC3	26	4	5	2	1	0	1
ABCC4	18	43	9	8	21	17	14
ABCC5	0	13.253	0	2	5	1	2
ABCC5-AS1	1	0	0	0	0	0	0
ABCC6	0	5.27	1.229	0	1	7	0
ABCC6P1	0	2.73	3.771	1	2	0	1

Figure 28. Example of a gene matrix with intron counts file.

4. Transcript Matrix Files

The transcript matrix file contains transcript counts for each barcode, with the transcripts in the rows and barcodes in the columns. The file contains raw counts that can be normalized and transformed using CogentDS.

	AAGAGCGCAACTTAAC	AAGAGCGCACCGAATT	AAGAGCGCAGAAGTAA	AAGAGCGCAGACCGTT	AAGAGCGCAGATTCAT	AAGAGCGCCAATCTTG
CCNL2-205	17	0	0	0	0	18.327
CCNL2-217	0	0	0	0	0	0
CCNL2-213	0	0	0	0	0	0
CCNL2-212	0	0	0	0	0	0
CCNL2-203	0	0	0	0	0	1.673
CCNL2-219	0	0	0	0	0	0
CCNL2-215	0	0	0	0	0	0
CCNL2-216	0	0	0	0	0	0
CCNL2-207	0	0	0	0	0	0
CCNL2-211	0	0	0	0	0	0
CCNL2-208	0	0	0	0	0	0
CCNL2-202	0	0	0	0	0	0
CCNL2-209	0	0	0	0	0	0
CCNL2-204	0	0	0	0	0	0
CCNL2-206	0	0	0	0	0	0
CCNL2-218	0	0	0	0	0	0
MRPL20-AS1-202	0	0	0	0	0	0
MRPL20-AS1-204	0	0	0	0	0	0
MRPL20-AS1-207	0	0	0	0	0	0
MRPL20-AS1-209	0	0	0	0	0	0
MRPL20-AS1-210	0	0	0	0	0	0
MRPL20-AS1-201	0	0	0	0	0	0
MRPL20-AS1-211	0	0	0	0	0	0
MRPL20-AS1-212	0	0	0	0	0	0
MRPL20-AS1-208	0	0	0	0	0	0
MRPL20-AS1-206	0	0	0	0	1	0
MRPL20-AS1-213	0	0	0	0	0	0
MRPL20-AS1-203	0	0	0	0	0	0
MRPL20-AS1-214	0	0	0	0	0	0
MRPL20-AS1-205	27.009	0	0	0	0	0
MRPL20-201	1.991	0	0	0	0	0

Figure 29. Example of a transcript matrix file.

5. Data Output from the SMART-Seq mRNA LP (with UMIs) Kit

SMART-Seq mRNA LP (with UMIs) is a hybrid kit that generates two types of sequencing reads:

- 5′ end reads containing UMIs
- 'Internal' reads that do not contain UMIs.

CogentAP analyzes both types of reads at the same time and then generates two types of result files for gene expression analysis.

- Rdata object file, gene matrix, transcript matrix and stats files calculated using only the 5′ UMI reads, to be used for UMI-based analyses. These files include the keyword `5pUMI` in the file names.
- Rdata object file, gene matrix, transcript matrix and stats file calculated with all reads, i.e., both 5′ UMI reads and internal reads, to be used for UMI-agnostic analyses. These files include the keyword "all" in the file names.

NOTE: No result files are calculated with just internal reads.

6. Gene and Transcript Info Files

The gene info file contains the main annotations for the genes as described in the GTF file that is part of the genome build.

Table 13. Columns in the `gene_info.csv` output file.

Column name	Description
Gene_ID	Gene ID used in CogentDS, typically the Ensembl ID
Gene_Name	The gene symbol
Gene_Biotype	The gene classification
Gene_Length	The gene length, used for some normalizations.

An example file screenshot is shown below.

Gene_ID	Gene_Name	Gene_Biotype	Gene_Length
ENSG00000228037	ENSG00000228037	lncRNA	2974
ENSG00000142611	PRDM16	protein_coding	369454
ENSG00000284616	ENSG00000284616	lncRNA	5467
ENSG00000157911	PEX10	protein_coding	9834
ENSG00000260972	ENSG00000260972	lncRNA	1697
ENSG00000224340	RPL21P21	processed_pseudogene	337
ENSG00000226374	LINC01345	lncRNA	4478
ENSG00000229280	EEF1DP6	processed_pseudogene	372
ENSG00000142655	PEX14	protein_coding	158471
ENSG00000232596	LINC01646	lncRNA	22536
ENSG00000235054	LINC01777	lncRNA	12663
ENSG00000231510	LINC02782	lncRNA	4441
ENSG00000149527	PLCH2	protein_coding	79553
ENSG00000284739	ENSG00000284739	lncRNA	9345
ENSG00000171621	SPSB1	protein_coding	76639
ENSG00000272235	ENSG00000272235	lncRNA	3461
ENSG00000284694	ENSG00000284694	lncRNA	5245
ENSG00000224387	ENSG00000224387	lncRNA	959
ENSG00000142583	SLC2A5	protein_coding	53373
ENSG00000284674	LINC02781	lncRNA	8188

Figure 30. Example of a gene info file. The gene length column denotes the length of the gene from the start to the end including introns.

The transcript info file contains the main annotation for the transcripts as described in the GTF file that is part of the genome build. This file has a similar format to gene info file. In the case for transcript info file, both gene ID and transcript ID are included in the file.

Table 14. Columns in the `transcript_info.csv` output file.

Column name	Description
Transcript_ID	Transcript ID used in CogentDS, typically the Ensembl ID
Transcript_Name	The transcript symbol
Gene_ID	Gene ID that the transcript is derived from.
Gene_Name	The gene symbol that the transcript is derived from.
Transcript_Biotype	The transcript classification
Transcript_Length	The transcript length, used for some normalizations.

Transcript_ID	Transcript_Name	Gene_ID	Gene_Name	Transcript_Biotype	Transcript_Length
ENST00000424215	ENST00000424215	ENSG00000228037	ENSG00000228037	lncRNA	2974
ENST00000511072	PRDM16-206	ENSG00000142611	PRDM16	protein_coding	365175
ENST00000607632	PRDM16-210	ENSG00000142611	PRDM16	retained_intron	117409
ENST00000378391	PRDM16-203	ENSG00000142611	PRDM16	protein_coding	366225
ENST00000514189	PRDM16-208	ENSG00000142611	PRDM16	protein_coding	365132
ENST00000270722	PRDM16-201	ENSG00000142611	PRDM16	protein_coding	369419
ENST00000512462	PRDM16-207	ENSG00000142611	PRDM16	protein_coding_CDS_not_defined	195995
ENST00000463591	PRDM16-204	ENSG00000142611	PRDM16	protein_coding	142787
ENST00000509860	PRDM16-205	ENSG00000142611	PRDM16	protein_coding	37803
ENST00000378389	PRDM16-202	ENSG00000142611	PRDM16	protein_coding_CDS_not_defined	9177
ENST00000606170	PRDM16-209	ENSG00000142611	PRDM16	retained_intron	857
ENST00000641871	ENST00000641871	ENSG00000284616	ENSG00000284616	lncRNA	5467
ENST00000288774	PEX10-201	ENSG00000157911	PEX10	protein_coding	8601
ENST00000447513	PEX10-202	ENSG00000157911	PEX10	protein_coding	8591
ENST00000650293	PEX10-209	ENSG00000157911	PEX10	nonsense_mediated_decay	8396
ENST00000507596	PEX10-204	ENSG00000157911	PEX10	protein_coding	7703
ENST00000510434	PEX10-206	ENSG00000157911	PEX10	nonsense_mediated_decay	6018
ENST00000508384	PEX10-205	ENSG00000157911	PEX10	protein_coding	5268
ENST00000515760	PEX10-208	ENSG00000157911	PEX10	protein_coding_CDS_not_defined	1422

Figure 31. Example of a transcript info file. The transcript length column denotes the length of the transcript from the start to the end of the genomic coordinates including introns.

B. Gene Fusion Files

NOTE: These files are only generated when the gene fusion option for RNA-seq analysis (Section V.B.2) is used for kits excluding Shasta Total RNA-Seq Kit, and for analysis with ≤5,000 barcodes.

Table 15. Raw data output files generated by CogentAP fusion analysis. Files can be found in the `count_matrices/` subfolder (for the csv files) and `fusion/` subfolder (for the gz files) of the output folder defined during the analysis run.

Referred to as	File name(s)
Junction matrix	`fusion_junctionCounts.csv`
Spanning matrix	`fusion_spanCounts.csv`
Junction Data	`junction/barcodes.tsv.gz`, `junction/features.tsv.gz`, `junction/matrix.mtx.gz`
Spanning Data	`span/barcodes.tsv.gz`, `span/features.tsv.gz`, `span/matrix.mtx.gz`

1. Junction matrix file

The junction matrix file contains junction read counts for each barcode. In the table, each gene fusion is a row, with the index barcodes (i.e., a cell) in the columns. The table values represent the number of reads detected tagged with the specified barcode that contain the corresponding fusion.

GeneFusion	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AAGGTCTG	AAGGTCTG	AAGGTCTG…
TPTE2P2--MALAT1	71	52	44	59	43	0	13	86	92	11
TPTE2P2--RPL37	13	9	5	9	13	0	8	5	33	4
MSH2--MRPS18A	9	0	0	0	0	0	0	0	0	0
TPTE2P2--PPIAP29	7	3	1	1	2	0	0	2	0	2
GPAT2--PFN1	9	0	0	1	1	0	0	0	0	1
WDR35--PFN1	7	0	0	0	5	0	2	1	7	2
RPS28--CHST2	5	0	0	0	0	0	0	0	0	0
TPTE2P2--B2M	7	1	10	2	13	7	4	1	13	4
CCNL1--YWHAQ	4	0	0	0	0	0	0	0	0	0
SEPTIN9--MALAT1	2	0	0	0	0	0	0	0	0	0
KNG1--RN7SL2	5	2	0	0	0	1	4	3	1	3
KNG1--AL627171.4	5	2	0	0	0	1	4	3	1	3
TPTE2P2--ACTG1	5	0	0	0	0	0	0	0	0	2
STAT1--CHSY1	5	0	0	0	0	0	0	0	0	0
PXN--ARAP2	3	0	0	0	0	0	0	0	0	0
ACTB--MYL12A	4	0	0	0	0	0	0	0	0	0
UCK2--MEAF6	3	0	0	0	0	0	0	0	0	0
AL135905.2--LMAN2	4	0	0	0	0	0	0	0	0	0
PLEK--OBSCN	6	0	0	0	0	0	0	0	0	0
TMEM59--GRK6	3	0	0	0	0	0	0	0	0	0
…										

Figure 32. Example of a junction matrix file.

2. Spanning matrix file

The spanning matrix file contains spanning read counts for each barcode, with the gene fusion in the rows and barcodes/cells in the columns. Each value is the number of paired-end reads containing the sequences of both genes that form the corresponding fusion.

GeneFusion	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AACCGGTT	AAGGTCTG	AAGGTCTG	AAGGTCTG…
TPTE2P2--MALAT1	0	0	0	0	0	0	0	0	0	0
TPTE2P2--RPL37	0	0	0	0	0	0	0	0	0	0
MSH2--MRPS18A	16	0	0	0	0	0	0	0	0	0
TPTE2P2--PPIAP29	0	0	0	0	0	0	0	0	0	0
GPAT2--PFN1	0	0	0	0	0	0	0	0	0	0
WDR35--PFN1	0	0	0	0	0	0	0	0	0	0
RPS28--CHST2	8	0	0	0	0	0	0	0	0	0
TPTE2P2--B2M	0	0	0	0	0	0	0	0	0	0
CCNL1--YWHAQ	6	0	0	0	0	0	0	0	0	0
SEPTIN9--MALAT1	14	0	0	0	0	0	0	0	0	0
KNG1--RN7SL2	0	0	0	0	0	0	0	0	0	0
KNG1--AL627171.4	0	0	0	0	0	0	0	0	0	0
TPTE2P2--ACTG1	0	0	0	0	0	0	0	0	0	0
STAT1--CHSY1	7	0	0	0	0	0	0	0	0	0
PXN--ARAP2	11	0	0	0	0	0	0	0	0	0
ACTB--MYL12A	5	0	0	0	0	0	0	0	0	0
UCK2--MEAF6	9	0	0	0	0	0	0	0	0	0
AL135905.2--LMAN2	0	0	0	0	0	0	0	0	0	0
PLEK--OBSCN	4	0	0	0	0	0	0	0	0	0
TMEM59--GRK6	4	0	0	0	0	0	0	0	0	0
…										

Figure 33. Example of a spanning matrix file.

3. Junction and Spanning Data files

These files are automatically used to generate fusion overlays in the final CogentDS Rdata file provided that the `--fusion` parameter was enabled during the full analysis (see Section VII.B.3). In the case of standalone fusion analysis, these files cannot be directly imported into CogentDS in the current version of the software but may be enabled in future versions.

C. Immune Profiling Files

NOTE: These files are only generated when the immune profiling option for RNA-seq analysis (Section V.B.3) is used.

Table 16. Raw data output files generated by CogentAP immune analysis. Files can be found in the `count_matrices/` subfolder of the output folder defined during the analysis run.

Referred to as	File name
Clonotype matrix	`immune_clonotype_matrix.csv`
Metadata	`immune_metadata.csv`
Summary	`immune_summary.csv`
Top 3 clonotype matrix	`immune_top3_clonotype_matrix.csv`
Top 3 metadata	`immune_top3_metadata.csv`
Top 3 summary	`immune_top3_summary.csv`

1. Clonotype Matrix

The clonotype matrix file contains clonotype counts for each barcode, with the clonotype by rows and barcodes (i.e., cells) in the columns. The file contains raw counts that can then be normalized and transformed using CogentDS.

The clonotype is defined as joining of V, D, and J genes, constant regions (C), and CDR3 amino acid (CDR3aa) sequences, connected by the dollar-sign (`$`) symbol.

```
<V gene>$<D gene>$<J gene>$<constant region><$CDR3 aa>
```

A period (`.`) is used in place of a segment that doesn't exist in the clonotype.

Examples:

```
TRBV20-1*01$TRBD2*02$TRBJZ-7*01$TRBC$CSAGSGRGGRAVEQYF

IGKV4-1*01$.$IGKJ4*01$IGKC$CQQYYSTPALTF
```

In the second clonotype, the D gene isn't present, so the period is used.

Table 17. Columns in the `*_clonotype_matrix.csv` output file.

Column name	Description
V-D-J-C-CDR3aa	The string of V, D, and J genes, constant region, and CDR3 amino acid segment details, concatenated by the $ symbol.
<barcode1> **…** **…** **…** **<barcode*N*>**	Subsequent columns correspond to the joint clonotype segments identified for the barcode listed in the column header. The cell values are a count of the clonotype reads found for the V-D-J-C-CDR3aa combination.

An example file screenshot is shown below.

V-D-J-C-CDR3aa	AATGGTAAT	CATAATGGT	CGCGGTCGT	CGCGGTCGT	TTGTAATAG	CGTAATGGT	CGAAGTCGT	CGTTGTCGT
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDAEISTGYYLPYYFDYW	0	3	0	0	0	3	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDEFSTGYYLPYYFDYW	0	5	0	0	0	5	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDEISTGYYIPYYFDYW	0	10	0	0	0	10	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDEISTGYYLPYYFDYW	0	35521	0	0	0	0	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDENSTGYYLPYYFDYW	0	3	0	0	0	3	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFGDEISTGYYLPYYFDYW	0	5	0	0	0	5	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVLDDEISTGYYLPYYFDYW	0	5	0	0	0	5	0	288
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CGRVFDDEISTGYYLPYYFDYW	0	7	0	0	0	7	0	0
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$RARVFDDEISTGYYLPYYFDYW	0	16	0	0	0	16	0	0
IGKV1-12*01$.$IGKJ5*01$IGKC$CQQASSFPPVTF	3	0	0	0	3	0	0	0
IGKV4-1*01$.$IGKJ4*01$IGKC$CQQYYSTPALTF	0	0	0	166	0	0	0	0
IGKV4-1*01$.$IGKJ4*01$IGKC$CQ_YYSTPALTF	0	0	0	4	0	0	0	3
IGKV4-1*01$.$IGKJ4*01$IGKC$RQQYYSTPALTF	0	0	0	3	0	0	0	0
IGKV4-1*01$.$IGKJ4*01$IGKC$SQQYNSTPALTF	0	0	0	2	0	0	0	1
IGLV3-19*01$.$IGLJ2*01$IGLC$CNSRDTSDNHLVF	0	7731	0	0	0	532	0	0
TRAV8-2*01$.$TRAJ12*01$.$CVVSPMASSYKLIF	0	0	2	0	0	0	2	0
TRAV8-2*01$.$TRAJ12*01$TRAC$CVVSPMDSSYKLIF	0	0	5	0	0	0	5	0
TRBV20-1*01$TRBD2*02$TRBJ2-7*01$TRBC$CSAGSGRGGRAVEQYF	0	0	22	0	0	0	22	0

Figure 34. Example of a clonotype matrix file.

2. Top3 Clonotype Matrix

The top3 clonotype matrix file is identical in format to the clonotype matrix file, but only contains information based on the three clonotypes identified with the maximum reads in the dataset, which correspond to the top3 rows of an intermediate raw clonotype file.

3. Metadata

The metadata file contains clonotypes and their V, D, J, and C segment details, corresponding CDR3 amino acid sequences, and two columns with a boolean value 'Y' or 'N' to mark if the clonotype is a light or heavy chain. Similar to the clonotype_matrix file above, the clonotype is defined as joining of V, D, and J genes, constant region (C), and the CDR3 amino acid sequences (CDR3aa), connected by the $ symbol.

Table 18. Columns in the `*_metadata.csv` output file.

Column name	Description
V-D-J-C-CDR3aa	The string of V, D, and J genes, constant region, and CDR3 amino acid segment details, concatenated by the $ symbol.
V	V segment of the clonotype.
D	D segment of the clonotype.
J	J segment of the clonotype.
C	C segment of the clonotype.
CDR3aa	CDR3 amino acid segment of the clonotype.
Light Chain	Boolean value (Y or N). A 'Y' value designates the clonotype as a light chain.
Heavy Chain	Boolean value (Y or N). A 'Y' value designates the clonotype as a heavy chain.

An example file screenshot is shown below.

V-D-J-C-CDR3aa	V	D	J	C	CDR3aa	Light Chain	Heavy Chain
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDAEISTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVFDAEISTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDEFSTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVFDDEFSTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDEISTGYYIPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVFDDEISTGYYIPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDEISTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVFDDEISTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFDDENSTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVFDDENSTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVFGDEISTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVFGDEISTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CARVLDDEISTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CARVLDDEISTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$CGRVFDDEISTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	CGRVFDDEISTGYYLPYYFDYW	N	Y
IGHV4-61*01$IGHD3-9*01$IGHJ4*02$IGHG1$RARVFDDEISTGYYLPYYFDYW	IGHV4-61*01	IGHD3-9*01	IGHJ4*02	IGHG1	RARVFDDEISTGYYLPYYFDYW	N	Y
IGKV1-12*01$.$IGKJ5*01$IGKC$CQQASSFPPVTF	IGKV1-12*01	.	IGKJ5*01	IGKC	CQQASSFPPVTF	Y	N
IGKV4-1*01$.$IGKJ4*01$IGKC$CQQYYSTPALTF	IGKV4-1*01	.	IGKJ4*01	IGKC	CQQYYSTPALTF	Y	N
IGKV4-1*01$.$IGKJ4*01$IGKC$CQ_YYSTPALTF	IGKV4-1*01	.	IGKJ4*01	IGKC	CQ_YYSTPALTF	Y	N
IGKV4-1*01$.$IGKJ4*01$IGKC$RQQYYSTPALTF	IGKV4-1*01	.	IGKJ4*01	IGKC	RQQYYSTPALTF	Y	N
IGKV4-1*01$.$IGKJ4*01$IGKC$SQQYNSTPALTF	IGKV4-1*01	.	IGKJ4*01	IGKC	SQQYNSTPALTF	Y	N
IGLV3-19*01$.$IGLJ2*01$IGLC$CNSRDTSDNHLVF	IGLV3-19*01	.	IGLJ2*01	IGLC	CNSRDTSDNHLVF	Y	N
TRAV8-2*01$.$TRAJ12*01$.$CVVSPMASSYKLIF	TRAV8-2*01	.	TRAJ12*01	.	CVVSPMASSYKLIF	Y	N
TRAV8-2*01$.$TRAJ12*01$TRAC$CVVSPMDSSYKLIF	TRAV8-2*01	.	TRAJ12*01	TRAC	CVVSPMDSSYKLIF	Y	N
TRBV20-1*01$TRBD2*02$TRBJ2-7*01$TRBC$CSAGSGRGGRAVEQYF	TRBV20-1*01	TRBD2*02	TRBJ2-7*01	TRBC	CSAGSGRGGRAVEQYF	N	Y

Figure 35. Example of a clonotype metadata file.

4. Top3 Metadata

The top3 metadata file is identical in format to the metadata file, but only contains information based on the three clonotypes identified with the maximum reads in the dataset, which correspond to the top3 lines of raw clonotype output file.

5. Summary

The summary file contains barcodes, total reads, and clonotypes identified per barcode. It has 62 columns, which include four components: barcode, total reads, clonotype category marks, and segment call details.

- The first two columns in the table are the barcode and total reads.
- Columns 3–14 are clonotype categories, which mark and summarize if a clonotype category was identified for the barcode, i.e.,
  - (For TCR): TRA, TRB, TRD, and TRG
  - (For BCR): IGG, IGD, IGA, IGM, IGE, IGH, IGK, and IGL

If a category is not identified, the cell value is left blank.

- Columns 15–62 are listed V, D, and J genes and constant region details. For example: IGHV-1, IGHD3-1, IGHJ4 and IGHG1 for the IGH category (i.e., four columns per clonotype category).

Table 19. Columns in the `*_full_summary.csv` output file.

Row #	Column name	Description
1	Barcode	The barcode sequence of a single cell.
2	Total_Reads	Number of reads with clonotype type identified for the corresponding barcode.
3–6	TRA, TRB, TRD, TRG	Categories for T-cell (TCR) chain types. The cell value will match the column header if the chain type is detected with the given barcode. If it is not detected, the cell will be blank.
7–14	IGG, IGD, IGA, IGM, IGE, IGH, IGK, IGL	Categories for B-cell (BCR) chain types. The cell value will match the column header if the chain type is detected with the given barcode. If it is not detected, the cell will be blank.

Row #	Column name	Description
15–18	TRA_V, TRA_D, TRA_J, TRA_C	(TCR) V, D, and J genes and constant region identified within the TRA chain type.
19–22	TRB_V, TRB_D, TRB_J, TRB_C	(TCR) V, D, and J genes and constant region identified within the TRB chain type.
23–26	TRD_V, TRD_D, TRD_J, TRD_C	(TCR) V, D, and J genes and constant region identified within the TRD chain type.
27–30	TRG_V, TRG_D, TRG_J, TRG_C	(TCR) V, D, and J genes and constant region identified within the TRG chain type.
31–34	IGG_V, IGG_D, IGG_J, IGG_C	(BCR) V, D, and J genes and constant region identified within the IgG chain type.
35–38	IGD_V, IGD_D, IGD_J, IGD_C	(BCR) V, D, and J genes and constant region identified within the IgD chain type.
39–42	IGA_V, IGA_D, IGA_J, IGA_C	(BCR) V, D, and J genes and constant region identified within the IgA chain type.
43–46	IGM_V, IGM_D, IGM_J, IGM_C	(BCR) V, D, and J genes and constant region identified within the IgM chain type.
47–50	IGE_V, IGE_D, IGE_J, IGE_C	(BCR) V, D, and J genes and constant region identified within the IgE chain type.
51–54	IGH_V, IGH_D, IGH_J, IGH_C	(BCR) V, D, and J genes and constant region identified within the IgH chain type.
55–58	IGK_V, IGK_D, IGK_J, IGK_C	(BCR) V, D, and J genes and constant region identified within the IgK chain type.
59–62	IGL_V, IGL_D, IGL_J, IGL_C	(BCR) V, D, and J genes and constant region identified within the IgL chain type.

6. Top3 Summary

The top3 summary file is identical in format to the summary file, but only contains information based on the three clonotypes identified with the maximum reads in the dataset, which correspond to the top3 lines of raw clonotype output file.

Appendix B. Analysis of Raw DNA-seq Data Files

A. Default Analysis Files

Table 20. Processed data output files generated by the default CogentAP analysis command for DNA-seq analysis.

Referred to as	File name
Ginkgo Data	`data`
Ginkgo SegCopy	`SegCopy`
Ginkgo SegFixed	`SegFixed`
Ginkgo SegNorm	`SegNorm`
Ginkgo SegStats	`SegStats`
Multiqc General Stats	`multiqc_general_stats.txt`
Multiqc FastQC results	`multiqc_fastqc.txt`

Referred to as	File name
Multiqc Alignment Stats	`multiqc_bowtie2.txt`
Multiqc Trimming Stats	`multiqc_trimmomatic.txt`
Multiqc Picard MarkDuplicates Stats	`multiqc_picard_dups.txt`
Multiqc Picard InsertSize Stats	`multiqc_picard_insertSize.txt`

1. Ginkgo Output Files

The output files from Ginkgo—`data`, `SegCopy`, `SegFixed`, `SegNorm` and `SegStats`—contain various metrics from the single-cell CNV analysis performed.

- Ginkgo Data—a tab-delimited file containing the raw read counts per bin per cell.
- Ginkgo SegCopy—a tab-delimited file containing final copy number estimates per bin per cell.
- Ginkgo SegFixed—a tab-delimited file containing read counts per bin per cell after segmentation, but before ploidy adjustment.
- Ginkgo SegNorm—a tab-delimited file containing read counts per bin per cell after GC-lowess normalization.
- Ginkgo SegStats—a tab-delimited file containing basic stats on read counts per bin for each cell.

2. Multiqc Output Files

The multiqc output files are the raw data files used to generate the tables and figures within the QC Metrics Report (Section VII.B.2).

- Multiqc General Stats—a tab-delimited file containing an overview of key values, taken from all the modules that were used in the analysis for each cell.
- Multiqc FastQC results—a tab-delimited file containing sequencing quality metrics for each cell.
- Multiqc Alignment Stats—a tab-delimited file containing alignment statistics derived from Bowtie2 for each cell.
- Multiqc Trimming Stats—a tab-delimited file containing trimming statistics derived from Trimmomatic for each cell.
- Multiqc Picard MarkDuplicates Stats—a tab-delimited file containing statistics derived from Picard MarkDuplicates for each cell.
- Multiqc Picard InsertSize Stats—a tab-delimited file containing statistics derived from Picard InsertSize metrics for each cell.

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

Notice to Purchaser

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

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

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