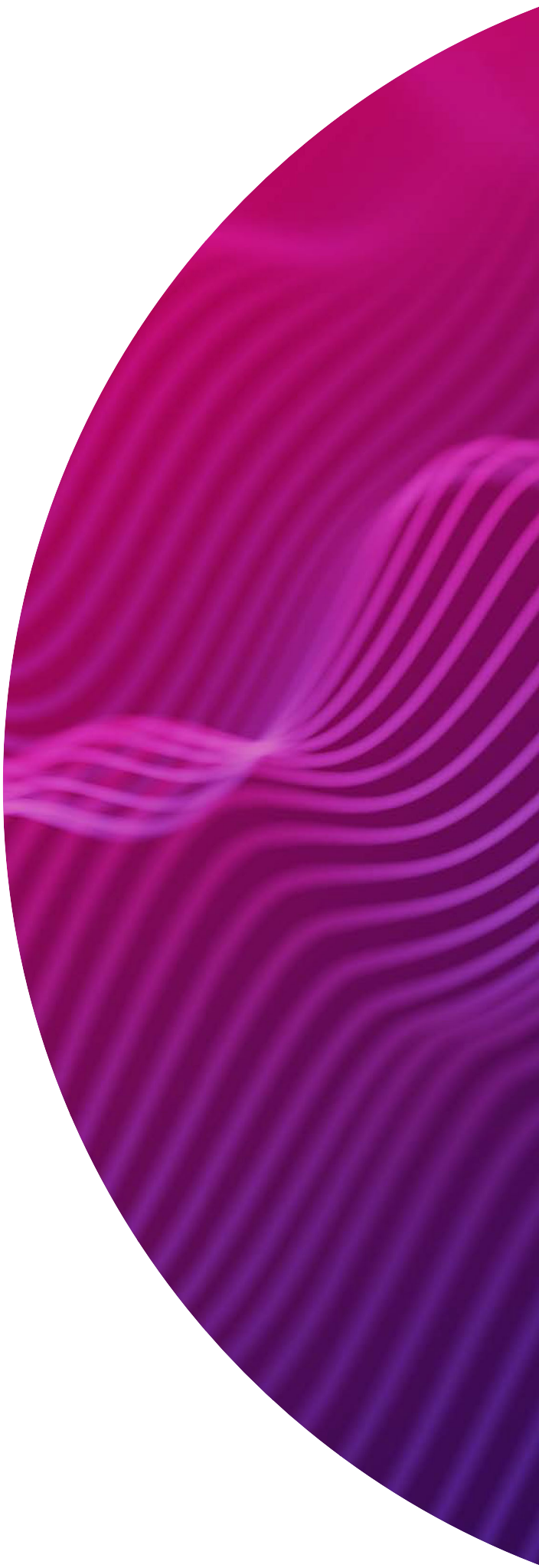
PacBio

Obc2fastq reference guide (v6.0)



Research use only. Not for use in diagnostic procedures.

P/N 103-393-900 Version 01 (March 2024)

© 2024 Pacific Biosciences of California, Inc. (“PacBio”)

Information in this document is subject to change without notice. PacBio assumes no responsibility for any errors or omissions in this document.

Certain notices, terms, conditions and/or use restrictions may pertain to your use of PacBio products and/or third party products. Refer to the applicable PacBio terms and conditions of sale and to the applicable license terms at https://www.pacb.com/legal-and-trademarks/product-license-and-use-restrictions/.

Trademarks:
Pacific Biosciences, the PacBio logo, PacBio, Circulomics, Omniome, SMRT, SMRTbell, Iso-Seq, Sequel, Nanobind, SBB, Revio, Onso, Apton, Kinnex and PureTarget are trademarks of PacBio.

PacBio
1305 O’Brien Drive
Menlo Park, CA 94025
www.pacb.com

Introduction

The `obc2fastq` utility is a command-line software tool that converts OBC (Onso™ Base Call) files generated by PacBio® Onso sequencers into FASTQ files. The utility extracts read sequences and quality scores.

Optionally, `obc2fastq` can perform sample demultiplexing if a sample sheet is provided. The sample sheet file contains the mappings between individual samples and the index barcode sequences. See "Sample sheet" on page 10 for details.

The `obc2fastq` utility is typically used by bioinformaticians, data analysts, or any researchers who handle PacBio Onso sequencing data for downstream analysis.

`obc2fastq` can also be run automatically on the Onso instrument computer **after** sequencing completes by selecting the **Enable FASTQ generation** option during run setup and uploading a sample sheet file. If the uploaded sample sheet contains indexes, demultiplexing will **also** be performed in addition to FASTQ generation. For more information, see **Onso™ system cluster generator and short-read sequencer Operations Guide.** An example sample sheet that can be edited for run setup is available here.

Hardware and software requirements

Hardware requirements
• **PC with 64-bit AMD/Intel processor with at least 8 logical cores (4 physical)**
• **16 GB RAM minimum. Typical maximum RAM usage is about 9 GB.**
For a 1.37 M spots x 336 cycles x 550 tiles run with 96 SIDs per lane: • Using a PC with an AMD EPYC-7643 processor with 48 cores (96 logical) and 512 GB RAM: Uses **9 GB of RAM**, and takes **9 minutes and 21 seconds**. • Using a laptop with an Intel i7-8650 processor with 4 core (8 logical) and 16 GB RAM: Uses **9 GB of RAM** and takes **63 minutes**.
Software requirements
• **Centos 7, Centos 8, Rocky 8, or Ubuntu operating systems**
• **gcc version 5.0 or greater, with C++17 support**

Installation

`obc2fastq` packages are available here and can be installed using standard Linux tools:

- **Centos 7 package**: Use the `yum` tool to install.
- **Centos 8 package**: Use the `yum` tool to install.
- **Ubuntu package**: Use the `dpkg` tool to install.
- **Rocky 8 package**: Use the `dnf` tool to install.

Centos with root permission

Use this procedure to install `obc2fastq` on a Centos systems with root permission. Use the `yum` package manager, specifying the name of the `obc2fastq.rpm` file:

```
$ sudo yum install obc2fastq-6.0.0-Linux.rpm
```

Note that a default `controls.fasta` file will be installed in `/usr/share/obc2fastq/data/controls.fasta`.

Centos without root permission

Use this procedure to install `obc2fastq` and its dependencies into the home directory **without** administrator privileges.

1. Create a local install folder and a folder to hold local `.rpm` files.

   ```
   $ mkdir ~/centos ~/rpm
   ```

2. Use `yum` to download the `tbb.x86_64` package `.rpm` file to the local folder. Also copy the `obc2fastq` rpm file to this folder.

   ```
   $ cp obc2fastq-6.0.0-Linux.rpm ~/rpm
   $ yumdownloader --destdir ~/rpm --resolve tbb.x86_64
   ```

3. Extract the `.rpm` files. Files will be installed into the local install folder.

   ```
   $ cd ~/centos
   $ rpm2cpio ~/rpm/tbb-*.rpm | cpio -id
   $ rpm2cpio ~/rpm/obc2fastq-*.rpm | cpio -id
   ```

4. Edit the `~/.bashrc` file to update the `$PATH` and `$LD_LIBRARY_PATH` environment variables.

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/centos/usr/
   lib64
   export PATH=$PATH:$HOME/centos/usr/bin
   ```

5. Reload `~/.bashrc`.

   ```
   $ source ~/.bashrc
   ```

In the above example, the `controls.fasta` file will be in `~/centos/usr/share/obc2fastq/data/controls.fasta`.

Running the obc2fastq utility

General usage

```
$ obc2fastq [OPTIONS] --input=<folder path>
```

Sample command-line usage

```
obc2fastq --input=D:\runs\FB0037073-BCC --controlsfile=D:\fastas\controls.fasta
```

Options	Description
`--input=<folder path>`	(**Required**) Specifies the full path to the sequencing run folder.
`--output=<folder path>`	Specifies the full path to the output folder where output files will be written. **Note:** If this folder does **not** exist, the software will create it.
`--controlsfile=<file path>`	Specifies the full path to the run's control FASTA file. Although this option is optional, it **must** be specified if demultiplexing based on control reads is to be performed.
`--samplesheet=<file path>`	Specifies the full path to the run's sample sheet file; **see** "Sample sheet" on page 10 for details. **Note**: We **require** that you use **only** alphanumeric characters, dashes, and underscores for the sample sheet name. (Default = `<run_folder>/SampleSheetUsed.csv`)
`--designsheet=<file path>`	Specifies the full path to the run's design sheet file. **Note**: Settings in the `sampleSheet.csv` file (if provided) will **override** settings in the `obc2fastq_params` file. (Default = `<run_folder>/obc2fastq_params.csv`)
`--flowcellid=<flow cell ID>`	Specifies the unique barcode associated with the flow cell scanned and loaded on the instrument. The default is determined from the metadata XML file.
`--n-padding=<on|off>`	Specifies whether to pad the FASTQ output with `N`s as needed. (Default = `off`)
`--save-sample-tag=<on|off>`	Specifies whether to save the demultiplexed sample ID to the header in the FASTQ output files. (Default = `on`)
`--trim-lowscores=<on|off>`	Specifies whether to trim reads with low quality scores at the end immediately after being loaded. (Default = `on`)
`--trim-q=<N>`	Specifies the minimum quality score for `trim-lowscores` end-trimming. (Default = `26`)
`--trim-window=<N>`	Specifies the window size for `trim-lowscores` end-trimming. (Default = `6`)
`--threads=<N>`	Specifies the maximum number of concurrent threads used in processing. The default is determined by the system.
`--logutctime=<on|off>`	Specifies whether to use universal time for the timestamp. (Default = `off`)

obc2fastq input files

The `obc2fastq` utility requires the **entire** sequencing run folder as input. The folder can be compressed for sharing with researchers using the `obc2fastq` utility, or with PacBio Technical Support during troubleshooting. This folder includes the following:

- `Base_Calls` folder: Contains the primary analysis base calling outputs generated by primary analysis, including the actual base call and quality scores calculated for each spot across all sequencing cycles. Most of the files in this folder are binary files, and they typically **cannot** be opened directly.
- `Metrics` folder: Contains human-readable CSV (comma-separated value) files with detailed statistics calculated during primary analysis, used for assessing the quality of the sequencing run. These files can be opened using a text editor, spreadsheet application, or parsed using custom scripting. These files are also used by PacBio Technical Support for visualizing and troubleshooting sequencing run quality.
- `<FlowCellBarcode>_metadata.xml` file: Stores metadata about the sequencing run.
- `transferred.status` file: A marker file indicating that the data transfer of the run folder to its destination output location (such as a network drive) has completed.
- `SampleSheetUsed.csv` file: A copy of the sample sheet file used to perform demultiplexing and FASTQ generation using `obc2fastq`. If a sample sheet was **not** imported during run setup, then a sample sheet will be automatically generated by ICS (Instrument Control Software) using the sequencing settings selected during run setup.
- `obc2fastq_params.csv` file: Contains demultiplexing and FASTQ generation settings used by `obc2fastq`.

obc2fastq output files

- `Control_Library.fastq.gz`: The gzipped FASTQ file for the controls. One file for each lane/read combination. This is generated when `OutputControlFASTQ=TRUE`.
- `Control_Library_Metrics.csv`: The comma-separated metrics file for the controls. One file for each lane/read combination. This is generated when `OutputControlFASTQ=TRUE`.
- `Sample_Library<_sampleID>.fastq.gz`: The gzipped FASTQ file for the samples. One file for each lane/read combination.
- `Sample_Library_Metrics.csv`: The comma-separated metrics file for the samples. One file for each lane/read combination.
- `<FlowCellBarcode>_Logs\Analysis\{FlowCellBarcode}_obc2fastq.log`: The run log.
- `<FlowCellBarcode>_Logs\Analysis\Metrics\<FlowCellBarcode>_Control_Library_Metrics.csv`: The comma-separated metrics file for the controls. One for each lane/read combination. This file is a duplicate of the one in the root folder of the output directory,

- `<FlowCellBarcode>_Logs\Analysis\Metrics\<FlowCellBarcode>_Sample_Library_Metrics.csv`: The comma-separated metrics file for the samples. One for each lane/read combination. This file is a duplicate of the one in the root folder.

Output file naming convention

Controls file:
`<Flowcell_id>_<LaneSpec>_<ReadSpec>_Control_library.fastq.gz`

Samples file:
`<Flowcell_id>_<LaneSpec>_<ReadSpec>_Sample_Library_<Sample_id>.fastq.gz` where:
`<LaneSpec>`= `L01` or `L02`
`<ReadSpec>` = `Index1`, `Index2`, `Read1`, `Read2`.

Output file location

- The `obc2fastq` log file is located in the `Logs/Analysis` folder and is named `<Flowcell_id>_obc2fastq.log`.
- Generated metrics files are separated by lane and placed into the `Metrics/L01` and `Metrics/L02` folders, and are named as follows:
  - `<Flowcell_id>_<LaneSpec>_<ReadSpec>_Control_Library_Metrics.csv`
  - `<Flowcell_id>_<LaneSpec>_<ReadSpec>_Sample_Library_Metrics.csv`
  - `<Flowcell_id>_<LaneSpec>_<ReadSpec>_Sample_Library_Metrics_tabled.csv`
- In addition, **all** metrics files are copied to the `Logs/Analysis/Metrics` folder.

FASTQ file output format description

Sequence data are represented in FASTQ format, with each sequence represented by four lines of data:

- **Line 1**: Read ID (**Note**: The format of this line is dependent on the obc2fastq `--save-sample-tag` option.)

Single UMI, `--save-sample-tag=off`:

```
@<instrumentID>:<runID>:<flowcell>:<lane>:<swathtile>:<x>:<y>:
<UMI> <track>:<filtered>:<0>
```

Dual UMI, `--save-sample-tag=off`:

```
@<instrumentID>:<runID>:<flowcell>:<lane>:<swathtile>:<x>:<y>:
<UMI1>+<UMI2> <track>:<filtered>:<0>
```

Single UMI, `--save-sample-tag=on`:

```
@<instrumentID>:<runID>:<flowcell>:<lane>:<swathtile>:<x>:<y>:
<UMI1> <track>:<filtered>:<0>:<SampleID1>
```

Dual UMI, `--save-sample-tag=on`:

```
@<instrumentID>:<runID>:<flowcell>:<lane>:<swathtile>:<x>:<y>:
<UMI1>+<UMI2> <track>:<filtered>:<0>:<SampleID1>+<SampleID2>
```

- **Line 2**: Sequence data (such as `CCAGT`...)
- **Line 3**: Comment line, which always begins with a **plus sign** (`+`).
- **Line 4**: Quality score data, which are Phred-scale quality scores encoded in ASCII-33 characters.

Examples

Dual index, 2 UMI, `--save-sample-tag=on`, track 1:

```
@BB507:4321:LP0000000-H:1:01001:1419:28:AANNNN+AAAAAA 1:N:0:GATATACC+CAACTGTA
```

2 UMI, `--save-sample-tag=off`, track 2:

```
@BB507:4321:LP0000000-H:2:01001:639:10:AAAAAA+AAAAAA 2:N:0
```

Obc2fastq options affect the output formatting as follows:

- Use the `--n-padding=on` option to automatically pad the end of read sequences with `N`s to ensure that all sequence data records are the same length.
- Use the `--save-sample-tag=on` option to add the demultiplexed sample ID to **Line 1**, the Read ID line.

Sample sheet

The sample sheet is a file containing sample information about a given sequencing run; it contains the mappings between individual samples and the index barcode sequences.

- A sample sheet is **required** if using the demultiplexing feature of `obc2fastq`, but it is **not** required to run `obc2fastq`.
- If a sample sheet is **not** provided, then for each lane, `obc2fastq` will generate:
  - A FASTQ file for each read containing **all samples** in that read.
  - A FASTQ file for each read containing **all controls** found in that read.

Following is a description of the sample sheet format, along with the elements in the sample sheet. Examples of elements and their constituent data are included in table format, followed by the CSV representation of the same element and data. An example sample sheet that can be edited for run setup is available here.

Sample sheet format

The sample sheet is a comma-delimited text file (`.csv`) that consists of the following elements.

Note: We **require** that you use **only** alphanumeric characters, dashes, and underscores for the sample sheet name.

Sections - Sections represent a group of data and contain the following records:

- **Field labels** - Used to identify the specific values for each section.
- **Field values** - Each field value is tied to a field label and represents the sample-specific information that corresponds to a sequencing run and is to be filled in for a given sequencing experiment.

Sections are identified within brackets such that each section name precedes the data for that section.

Example: `[<SectionName>] section data …`

Allowed values for `SectionName` are: `Run Information`, `Flow Cell Settings`, `Obc2fastq Settings`, and `Samples`.

Note: Every sample sheet **must** include these four sections. Within a section, some settings are optional.

Run Information section (Required)

The **Run Information** section contains metadata about the run and can be used by downstream analysis.

1. **(Required) FileFormatVersion** –Currently `2`.
2. **(Required) InstrumentPlatform** – Currently `Onso`.
3. **(Optional) FlowCellBarcode** - The unique barcode associated with the flow cell that is scanned and loaded on the instrument.
4. **(Optional) RunName** – User-specified text string.

[Run Information]	
FileFormatVersion	2
InstrumentPlatform	Onso
FlowcellBarcode	FC123456
RunName	MyRunName

```
[Run Information]
FileFormatVersion,2
InstrumentPlatform,Onso
FlowcellBarcode,FC123456
RunName,MyRunName
```

Flow Cell Settings section (Required)

The **Flow Cell Settings** section mirrors the Instrument Control Software (ICS) settings used by the Onso instrument.

Allowed field labels and values:

1. **(Required) Read1Cycles** – Integer, specifies the number of cycles run for insert 1.
2. **(Required) Read2Cycles** – Integer, specifies the number of cycles run for insert 2.

Note: For the number of **Read1Cycles** and **Read2Cycles**, add **2** cycles to the desired run configuration. For example, 100 cycles should be entered as `102`; 150 cycles should be entered as `152`.

3. **(Required) Index1Cycles** –Integer, specifies the number of cycles run for index 1.
4. **(Required) Index2Cycles** – Integer, specifies the number of cycles run for index 2.
5. **(Required) CustomPrimer** - Must be `TRUE` or `FALSE`. Specifies if custom primers were used for the run.
6. **(Required) OBC2FASTQ** - Must be `TRUE` or `FALSE`. Specifies if a FASTQ file should be generated.

[Flow Cell Settings]	
Read1Cycles	152
Read2Cycles	152
Index1Cycles	8
Index2Cycles	8
CustomPrimer	FALSE
OBC2FASTQ	TRUE

Obc2fastq Settings section

The **Obc2fastq Settings** section include the settings supported by the `obc2fastq` software.

- All of the settings in this section are **optional**.
- For boolean settings, if the setting is included, it is set to `TRUE`. If the setting is **not** included or the value is not specified, it is set to its **default** value.

Allowed field labels and values:

1. **IncludeTiles** – Used to specify the exact tiles that are to be **included** in the data processing. If **not** specified, **all** tiles are included.
   Tiles are specified using the format `L<Lane>/S<Swath>_T<Tile>`.
   For example, to specify lane 2, swath 3, tile 15 use `L02/S03_T015`.
   A comma-separated list of tiles can be specified, but the list **must** be surrounded by quote marks.
   Example: `"L01/S02_T002,L02/S03_T015"`.
   Note: This setting **cannot** be used together with the **ExcludeTiles** setting.
2. **ExcludeTiles** – Used to specify which tiles to **exclude** from processing. If **not** specified, all tiles are **included**. The format used is the same as with **IncludeTiles.**
   Note: This setting **cannot** be used together with the **IncludeTiles** setting.
3. **OutputControlFASTQ** – If set to `TRUE`, FASTQ files for the control reads are generated. This is `TRUE` by default.
4. **I1Mismatches** – Can be `0`, `1`, or `2`; the default is `1`. Sets the maximum number of mismatches allowed in index 1 for performing demultiplexing.
5. **I2Mismatches** – Can be `0`, `1`, or `2`; the default is `1`. Sets the maximum number of mismatches allowed in index 2 for performing demultiplexing.
6. **OutputIndexFASTQ** – If set to `TRUE`, FASTQ files for index reads are generated. This is `FALSE` by default. **Note**: Reads in the I1/I2 FASTQ files should be written in the same order as the R1/R2 FASTQ files.

7. **MergeLanes** – If specified, R1 and R2 reads with the same `Sample_ID` in different lanes are merged together into the same R1 and R2 FASTQ files. If **not** specified, the reads are **not** merged.
8. **R1CycleUsage** – Specifies the cycle masks for R1. The masks used specify which cycles of the `.obc` data tracks (T1,T2,T3,T4) to pull data from for demultiplexing into the `Read 1` FASTQ output files. See "Appendix B - Cycle masks" on page 19 for more information.
9. **R2CycleUsage** – Specifies the cycle masks for R2. The masks used specify which cycles of the `.obc` data tracks (T1,T2,T3,T4) to pull data from for demultiplexing into `Read 2` FASTQ output files. See "Appendix B - Cycle masks" on page 19 for more information.
10. **I1CycleUsage** – Specifies the cycle masks for I1. The masks used specify where the cycles are located for Index 1 demultiplexing. See "Appendix B - Cycle masks" on page 19 for more information.
11. **I2CycleUsage** – Specifies the cycle masks for I2. The masks used specify where the cycles are located for Index 2 demultiplexing. See "Appendix B - Cycle masks" on page 19 for more information.
12. **U1CycleUsage** – Specifies the cycle masks to use for UMI data. See "Appendix B - Cycle masks" on page 19 for more information.
13. **U2CycleUsage** – Specifies the cycle masks to use for UMI data. See "Appendix B - Cycle masks" on page 19 for more information.
14. **TrimLowScores** - Must be `TRUE` or `FALSE`; default = `TRUE`. If `TRUE`, sequences are trimmed at the end immediately after being loaded in case Q-scores of bases on the right end fall below a certain threshold. Assuming a threshold (`T`), sequences are scanned from right to left through a window size of (`M`) bases. If **all** bases within the window have a Q-score >= `T`, the trim point is set to be end of the window. This means that If **all** (`M`) bases at the end of the read have a Q-score >= `T`, **no** trimming is performed.
15. **TrimQ** - (Default = `26`) Specifies the Q-score threshold (`T`) used for the end-trimming in the **TrimLowScores** setting.
16. **TrimWindow** - (Default = `6`) Specifies the window size used for trimming the ends of bases (`M`) in the **TrimLowScores** setting.

[Obc2fastq Settings]	
IncludeTiles	"L01/S01_T001,L01/S01_T002"
OutputControlFASTQ	TRUE
I1Mismatches	1
I2Mismatches	1
OutputIndexFASTQ	FALSE
MergeLanes	FALSE
TrimLowScores	FALSE
TrimQ	26
TrimWindow	6

```
[Obc2fastq Settings]
IncludeTiles,"L01/S01_T001,L01/S01_T002"
OutputControlFASTQ,TRUE
I1Mismatches,1
I2Mismatches,1
OutputIndexFASTQ,FALSE
MergeLanes,FALSE
TrimLowScores,FALSE
TrimQ, 26
TrimWindow,6
```

Samples section

The **Samples** section includes sample information and specifies whether or not demultiplexing is performed if the sample number 2 or more.

Note: The minimum entries required for the [**Samples**] section are the `[Samples]` and the `Biosample`, `Lane`, `Index`, and `Index2` lines**.** All headings are **required**.

- The `BioSample` field contains the sample name which can be named using any printable ASCII characters **except** spaces.
- `Index` and `Index2` are the index barcode sequences used to demultiplex samples that are on the same lane. `Index` and `Index2` sequences must be `[ACTG]`.
- The `Lane` field must be either `1` or `2`.
- For Onso indexed adapters, the Index P sequence goes in the `Index` field, and the Index A sequence goes in the `Index2` field.
- For Onso conversion libraries, the i7 sequence goes in the `Index` field, and the i5 index sequences goes in the `Index2` field.

[Samples]			
Biosample	Lane	Index	Index2
BioSample1	1	TCCTTAGG	CAACTGTA
BioSample2	1	CGTCGCAC	GCGTCACT
BioSample2	2	GAGAAGCT	TGCAACGG
BioSample3	2	ATAGCTTA	ATTGGTAC
BioSample4	2	CTCAGACA	GAGCATCA

```
[Samples]
Biosample,Lane,Index,Index2
BioSample1,1,TCCTTAGG,CAACTGTA
BioSample2,1,CGTCGCAC,GCGTCACT
BioSample2,2,GAGAAGCT,TGCAACGG
BioSample3,2,ATAGCTTA,ATTGGTAC
BioSample4,2,CTCAGACA,GAGCATCA
```

Appendix A - Error messages

`Could not read <description>: '<filename>'`
: The file is not found or is not readable.

`Could not create folder '<folder>'`
: The folder could not be created. Verify the name of the folder and permissions.

`Could not create file '<file>'`
: The file could not be created using the specified path. Verify the file and folder permissions.

`Could not find folder: "<folder>"`
: The specified folder could not be located.

`Expecting name of folder: "<folder>"`
: A folder was expected, but a file was found instead.

`Cannot copy file "<filename>" from "<filepath>" to "<folder>"`
: This error might occur if a folder in the output path is not writable, or the specified file name is not writable.

`Metadata file '<filename>' not loaded from '<run folder>':`
: An XML metadata (`.xml`) file is required to run `obc2fastq`. The file is normally generated as part of a run. The file name is formatted as `<FlowcellId>_metadata.xml`. If this file is **not** found, `obc2fastq` will try to load the file `metadata.xml`.

`Unable to locate any input files in '<run folder>'`
: `Obc2fastq` will search under the folder specified by the `--input` argument for subfolders containing `.obc` files. If **no** `.obc` files are located, this error is generated.

`Missing .obc files in "<folder>"`
: `Obc2fastq` located a file folder, but the folder was empty.

`Inconsistent number of .obc files in tile folder: "<folder>"`
: `Obc2fastq` located a tile folder, but one or more of the expected `.obc` files was not present.

`Inconsistent number of position files in tile folder "<folder>"`
: `Obc2fastq` requires `RN_positions` files to be located with `.obc` files. These files are also generated by primary analysis.

`Inconsistent number of spots in .obc file: "<filename>"`
: Each `.obc` file for each read in a tile folder must contain the same number of spots. This error could occur if a `.obc` file from a different tile folder was copied into the wrong folder.

`Inconsistent number of cycles in .obc file: "<filename>"`

All `.obc` files for a particular read must contain the same number of cycles. This error could occur if a `.obc` file from a different tile folder was copied into the wrong folder.

`File check failed`

Some input files may be missing or unreadable. Input files may not be in their expected location.

`<file_description> not specified.`
`[IncludeTiles,ExcludeTiles] Invalid tile folder name: '<folder name>'`

The IncludeTiles/ExcludeTiles field in the `[Obc2fastq Settings]` section of the sample sheet does not have the correct format.

`Could not open log file '"<filename>"`

The `obc2fastq` log file could not be created. Verify that the folder specified by the `--output` option is writable. If the `--output` option is **not** specified, then the folder specified by `--input` needs to be writable.

`Cannot open metrics file for output: "<filename>"`

Could not create a metrics file under either the `Metrics` folder or the `Logs/Analysis/Metrics` folder. Verify that the path shown in the message exists and has write permissions.

`Incorrect file type: "<filename>"`
`Incorrect file version: "<filename>"`

The `.obc` file shown in the message was corrupted or is in some way not loadable.

`Unable to open controls (fasta) file: <filename>`

Verity that the file specified by the `--controlsfile` option is present and readable.

`[Index,Read] invalid cycle range: <start>-<stop>`

The range of cycles specified by the cycle mask in the sample sheet is not valid.

`[Index,Read]: attempt to use cycles <start>-<stop> for gram T<track>. <cycles> available.`

The number of cycles actually found in the run did not match up to the cycle range specified by the corresponding cycle mask in the sample sheet. Verify that the cycle mask is correct with respect to the total run cycles.

`[Index,Read]: Specified track T<track> is out of range.`

The number of reads and index specified in the run configuration does not match up with a cycle mask `track` specifier. Verify that the cycle mask is correct with respect to the total number of read tracks. For example, one of the cycle masks specifies `T3:Y*` for a single-ended single index run.

```
'<fasta file>': <line>: Invalid file format; expecting sequence
'<fasta file>': <line>: Invalid file format; expecting '>'
```

The `.fasta` file specified by the `--controlsfile` option is invalid.

Appendix B - Cycle masks

A **cycle mask** specifies a set of cycles for a demultiplexing operation. Within a cycle mask, a series of operators indicates whether cycles are either **included** or **skipped**.

A positive integer or asterisk follows each operator to indicate a count of how many cycles are referenced.

- A `Y` (yes) operator indicates that a cycle is to be **used**.
- A `N` (no) operator indicates that a cycle is to be **skipped**.
- A positive integer indicates the number of cycles to include or exclude.
- An **asterisk** functions as a wild card, matching any remaining cycles in the read.

Examples:

- `Y4N*` - Indicates that only the first four cycles are to be used.
- `N3Y2N*` - Skips the first three cycles, uses the fourth and fifth cycles, and skips the remaining cycles.

Track identifiers

A cycle mask begins with a **track identifier** that specifies one of the `.obc` files produced by the base caller. Depending on the sequencing run, there can be between 1 and 4 files produced, such as `R1.obc`, `R2.obc`, `R3.obc`, and `R4.obc`. Each track identifier is followed by a colon (such as `T3:`).

- Example cycle mask that references the first 50 cycles of track 4 (`R4.obc`) and skips the reset of the cycles: `T4:Y50N*`

Cycle lengths

A cycle mask must define the full cycle length of a read, regardless of whether you are masking select bases in the read or all bases. For example, if the Track 1 produced by `callbase` consists of 30 bases and you want to mask the first 15, end the base mask with the total number of cycles. The base mask `T1:Y15N15` masks the first 15 bases (`Y15`) of Track 1 (`T1:`) and leaves the remaining 15 bases unmasked (`N15`).

Alternatively, `T1:Y15N*` achieves the same goal, but uses an asterisk to cover the remaining number of cycles.

Example cycle masks

- `T1:Y2N*`– Matches the first two cycles of track 1.
- `T3:N3Y100N3` – Matches 100 cycles of track 3 skipping the first and last 3 cycles.
- `T3:N2Y*N2` – Matches all but the first two and last two cycles of track 3.

Use of cycle masks in the obc2fastq section

The **[Obc2fastq Settings]** section of the sample sheet uses cycle masks for settings `R1CycleUsage`, `R2CycleUsage`, `I1CycleUsage`, `I2CycleUsage`, `U1CycleUsage` and `U2CycleUsage`.

- The masks in `R1CycleUsage` and `R2CycleUsage` specify which cycles of the `.obc` data tracks (`T1`,`T2`,`T3`,`T4`) to pull data from for demultiplexing into the `Read 1` and `Read 2` FASTQ output files.
- The masks in `I1CycleUsage` and `I2CycleUsage` specify where the cycles are located for Index 1 and Index 2 demultiplexing.
- The masks in `U1CycleUsage` and `U2CycleUsage` specify the cycles to use for UMI data.

Appendix C - FASTA file usage

`Obc2fastq` can detect **both** control sequences and adapters. Both sequences are contained in a FASTA file specified using the `--controlsfile` option. If the `.obc` data is to be demultiplexed into "control" FASTQ files, then a FASTA file **must** be specified containing the exact control sequence (or sequences) used in library preparation.

The format of a FASTA file is simple. Each sequence is preceded by a header line which begins with a **greater than** symbol (>). The heading for a control sequence can optionally contain the control type to be used to identify the control sequence in the generated metrics files. The control type is preceded by `ControlClass:` or

```
>{Name}|ControlClass:{class}
```

For example, if the control type is `LQC`, a valid heading followed by its sequence could be:

```
>LQCv02_01|ControlClass:LQC
GGGCGGCGACCTCGCGGGTTTTCGCTATTTATGAAAATTTTCCGGTT...
```

The adapter sequences used in the run may also be specified in the FASTA file. For example:

```
>adapter_Maia_R2
ATCGATTCGTGCTCGATGAACCGGGCGCTTA
```

Appendix D - Obc2fastq flow diagram