Olink® Explore CLI

Technical Information

1329, 1.3.0, 2024-03-07

Table of contents

1. Introduction

Olink® Explore CLI is a command-line interface (cli) for the Olink® Explore product. The application is capable of performing normalization, quality control (QC) and CV computations on NGS data and exporting the results on several supported formats.

1.1 Olink® Explore Software terminology and documentation

For detailed descriptions of Olink Explore software terminology and output files, refer to the Olink® NPX Explore Software User Manual, which shares the same Olink Explore software library for normalization, QC and output file generation.

Term	Description
NGS run folder	NGS sequencing output from a single flowcell.
Pre-processing	Conversion of NGS output to counts per Olink Explore index-barcode sequence.
Plate layout file	A csv-file containing sample id and sample type per well of one 96-well plate.
Run unit	The smallest set of 96-well plates and assays that can be run/re-run in the lab and analyzed in Olink Explore software. One counts file from the pre-processing contains counts for one run unit: i.e. one 96-well-plate and one Olink® Explore 3072 panel type.
Panelplate	A 96-well-plate run with one Olink Explore 3072 panel type, i.e. one Olink Explore 3072 run unit.
Project	A set of run units and all associated sample/panel lot/QC data, plus project metadata. Projects are, by design, independent of each other in Explore software. Project summary statistics (Inter-CV, CV distribution, Detectability per panel type) depend on included run units.

Output file	Description
Data file	A parquet-file with one record per combination of sample/control/external control and assay/internal control.
NPX™ file	A csv-file with one record per combination of sample/control and assay.
Extended NPX™ file	A csv-file with one record per combination of sample/control/external control and assay/internal control.
Analysis Report	A pdf-file with a summary of inter- and intra-CV per panel type, detectability and other project metrics.
CV-table file	A csv-file with a summary of inter- and intra-CV per block and panel type.

1.2 Requirements

The program has the following system requirements:

- A .Net 8 runtime installed on the system.
- A reasonably new Linux operating system. Supported distributions are Ubuntu (20.04, 22.04) and RHEL (8, 9). Most other modern Linux distributions should work as well, but are not tested by Olink.

2. Installation

To install the explore-cli program run the supplied installation script for your distribution:

- For **Ubuntu 20.04**: ./ubuntu-20.04-CLI-install.sh
- For **Ubuntu 22.04**: ./ubuntu-22.04-CLI-install.sh
- For **Red Hat 8+**: ./rehl8-CLI-install.sh

The installation scripts also installs the required .NET 8 runtime in addition to the application binary. When the installation is finished it should be possible to run the explore-cli command from anywhere.

Official support for installing the .NET 8 runtime for RHEL and Ubuntu can be found here:

- *RHEL*
- *Ubuntu*

2.1 Self-contained executable

There is a self-contained version of the CLI, which does not need the .NET 8 runtime to be installed. To install the self-contained version, give execute permission to the file and add to path if wished.

```
chmod +x explore-cli
cp explore-cli /usr/local/bin
```

2.2 Docker

The CLI is available as a Docker image for environments demanding it. Usage of Olink Explore CLI through Docker requires a more complex setup and is only recommended if installation of the required .NET runtime is not possible.

2.2.1 Installing the container image

Extract the zip file containing the CLI docker image and import it through `docker load:`

```
docker load -i explore-cli.tar
```

2.2.2 Running the container

The Explore CLI container is invoked through the `docker run` command:

```
docker run --rm explore-cli info
```

2.2.3 Running the container while bind mounting the host file system

The following command creates a new Olink Explore project in the current working directory with an json input file named `project.json`. Please note that relative/absolute file paths referenced in the json file must also exist within the bind mounted directory for them to be visible to the container. The user argument is necessary to keep the file ownership of the generated project file to the user executing the container.

```
docker run --rm \
--mount type=bind,source=”${PWD}”,target=/data \
-u $(id -u ${USER}):$(id -g ${USER}) \
explore-cli create -i /data/project.json -o /data/DockerProject
```

2.3 Logging

The program writes logs to std out that can be redirected to a file if need be. The log level can be controlled with the environment variable `OLINK_LOG_LEVEL` and has the following values where `warn` is default:

- trace
- debug
- info
- warn
- error

Should an unrecognized environment value be set the default log level `warn` will be used.

The log level can either be set in the users shell rc file (.bashrc for example) or set directly in the current shell:

```
OLINK_LOG_LEVEL=info explore-cli --help
```

2.4 Explore projects

The program performs operations on a group of one or more NGS runs called a project. The run units in a project are normalized and quality controlled together. By grouping related plates/run units together the software can ensure data is correct and changes are applied in a consistent manner.

The project data format is interchangeable with the Olink Explore desktop software (Olink NPX Explore) under the following conditions:

- The version of the software opening the project is greater than or equal to the one generating it.
- The version of the software opening the project is less than the one generating it but no breaking changes has been introduced between the versions. Breaking changes includes:
  - Updated reference values
  - Updated QC and normalization specification
  - Underlying project data format changes

2.4.1 Project data format

The structure of a project looks something like this:

```
.
├ checksum.txt
├ data
│   ├ 0599f530ca6b4293be699dd437e29779.gz
│   ├ 28885fb59b0049a0a8c90bafda6e62d9.gz
│   ├ 320518807f5945708ad1b1d4eadf69a5.gz
│   ├ 40b427846c444183ba16458432ca3a7c.gz
│   ├ 665a1d32f3794fdba1ab0229c23a456e.gz
│   ├ 9c012bb8d50b4fa08ecc8229433315f7.gz
│   ├ d23a32692c7240809cf781beea452fd9.gz
│   ├ efe0f80c44b2412b8b527aed1222b7f1.gz
└ project.explore
```

Directory	File description
checksum.txt	File containing checksum for ensuring validity of other project files.
data	Directory containing the compressed run units that have been added to the project.
project.explore	File containing the main project information.

The maximum supported amount of run units in a project is soft capped at 512 (64 full plates of all 8 panel types).

2.5 Verbs

The CLI is divided into several smaller functions called verbs. Each verb is responsible for performing a specific action in the Olink Explore workflow.

The available verbs are the following:

- info
- create
- export
- readme

2.5.1 Verb: info

Displays relevant information about the software.

Outputs a json formatted string with the following information:

Field	Description
versions.cli	The version of the CLI generating this information.
versions.exploreLibrary	The version of the underlying explore compute module.
versions.assayReferenceData	The version of the assay reference values specification.
versions.normalizationAndQcSpecification	The version of the specification for normalization and QC calculations.

versions.outputFileFormat	The version of the data output formats (NPX, extended NPX, cvTable and Analysis Report).
referenceValues.Cardiometabolic	The list of available lot numbers in this version of the software for the Cardiometabolic panel.
referenceValues.Inflammation	The list of available lot numbers in this version of the software for the Inflammation panel.
referenceValues.Neurology	The list of available lot numbers in this version of the software for the Neurology panel.
referenceValues.Oncology	The list of available lot numbers in this version of the software for the Oncology panel.
referenceValues.Cardiometabolic_II	The list of available lot numbers in this version of the software for the Cardiometabolic II panel.
referenceValues.Inflammation_II	The list of available lot numbers in this version of the software for the Inflammation II panel.
referenceValues.Neurology_II	The list of available lot numbers in this version of the software for the Neurology II panel.
referenceValues.Oncology_II	The list of available lot numbers in this version of the software for the Oncology II panel.

Example 1

```
explore-cli info
```

2.5.2 Verb: create

Creates a new Olink Explore project.

Short option	Long option	Required	Description
-o	--output	yes	Directory to create the project in, the last subfolder will be the directory containing project files. Subfolders that do not exist will be created. If last subfolder exists but is not empty the operation will fail with an error message.
-i	--input	yes	Input file of project properties to create project from. If not specified, an empty project will be created.
-n	--name	no	Name of project to be created. Takes precedence over project name in input file.
	--warnings-as-errors	no	True if warnings in the project definition should fail the operation, otherwise they will only be logged.

This verb creates a new explore project based on the contents of a specified json input file. An empty project can be created by leaving out the input file. The format and behavior of the input definition is explained below:

Relative paths in the input file are always resolved against the path of the input file itself.

Plate layout files are per default imported with plate id = file name without the .csv extension. If another plate id is desired the `plate id` field can be specified to override it. It is this plate id that must be used when referencing a specific plate layout in a run unit.

Index plate is a required property in the run unit definition and maps to the start and end sample number of a plate with 96 wells:

Start sample number	End sample number	Index plate
001	096	1
097	192	2
193	288	3
289	384	4

If running the operation with `warnings-as-errors` the operation will fail if the same plate layout has been connected to run units of different index plates within the same run. Otherwise a warning will be logged. Plate layouts connected to run units of different index plates between runs is considered OK.

Counts files are required to be in the same directory as their corresponding run metadata (run_metadata.json).

All panel used in the project must be mapped to one lot number in the dictionary `selectedLots`, and all run units in the project must be mapped to exactly one panel. Run units with explicit or estimated panels not included in selected lots will cause an error and abort the operation.

Panel is an optional field in the run unit definition. If none is provided the estimated panel, automatically detected by the pre-processing software and stored in the run metadata, will be used. If an estimated panel is not available and no panel has been specified the operation will fail. The operation will also fail if there is a mismatch between the specified panel and the estimated one.

Run units can be marked as either included or not included. If several run units are connected to the same panel and plate layout; only the first mentioned run unit will be imported as included.

Example json input format (contract version 1)

```
{
    “Version”: 1,
    ”projectName”: ”TestProject-CSAS3_EXPL”,
    ”productType”: ”Explore3072”,
    ”normalization”: ”Intensity”,
    ”sampleMatrix”: ”Blood plasma”,
    ”customerName”: ”Customer A”,
    ”customerEmail”: ”customer@company.com”,
    ”businessDevelopmentManagerName”: ”Manager A”,
    ”businessDevelopmentManagerEmail”: ”manager@company.com”,
    ”analysisLabName”: ”Lab A”,
    ”analysisLabEmail”: ”lab@company.com”,
    ”reportComment”: ”Comment for report”,
    ”annotations”: {
          ”key1”: ”value1”,
          ”key2”: ”value2”
    },
    ”selectedLots”: {
          ”Inflammation”: ”B14806”,
          ”Cardiometabolic_II”: ”B22605”
    },
    ”plateLayouts”: [
        {
            ”path”: ”plate_layouts/SS221117.csv”,
            ”plateID”: ”SS22117”,
        }
    ],
    ”runs”: [
        {
            ”path”: ”220511_A01243_0683_BH2NC5DRX2”,
            ”units”: [
                {
                    ”plateLayout”: ”SS221117”,
                    ”libraryNumber”: 1,
                    ”indexPlate”: ”1”,
                    ”panel”: ”Inflammation”,
                    ”included”: true
                }
            ]
        }
    ]
}
```

Field descriptions

Field	Description	Data type	Required	Comment
version	The version of the json input file contract	int	yes	Supported versions: 1
projectName	Name of the product used in project	string	yes	
productType	Type of the project	string	no	’Explore3072’ (default)
normalization	Project normalization setting	string	no	’PlateControl’ (default), ’Intensity’
sampleMatrix	Sample matrix type	string	no	
customerName	Project customer name	string	no	
customerEmail	Project customer email	string	no	Email adress
businessDevelopment ManagerName	Project business development manager	string	no	
businessDevelopment ManagerEmail	Project business development manager email	string	no	Email adress
analysisLabName	Analysis lab name	string	no	
analysisLabEmail	Analysis lab email	string	no	Email adress
reportComment	Comment text to be included in analysis report	string	no	
annotations	Dictionary of optional key value pairs	Dictionary<string,string>	no	

selectedLots	Dictionary of panel type to lot number selections	Dictionary<string,string>	yes	ex. 'Inflammation': 'B14806'
plateLayouts	Plate layouts to be included in the project	PlateLayout[]	yes	
plateLayouts[].path	File path of the plate layout file	string	yes	
plateLayouts[].plateId	Identifier of the plate layout		no	Will be set to the plate layout file name if not set
runs	Runs to be included in the project	Run[]	yes	
runs[].path	Path to the run folder		yes	Folder containing run_metadata.json and counts files
runs[].units	Units in the run to be included	RunUnit[]	yes	
runs[].units[].plateLayout	Plate layout of the run unit		yes	Must be the plateId of a plate layout defined under the plate layouts section
runs[].units[].libraryNumber	Library number of the run unit	int	yes	The same as lane number for Illumina instruments
runs[].units[].indexPlate	Index plate of the run unit		yes	'1', '2', '3', '4'
runs[].units[].panel	Supposed panel of the run unit		no	'Inflammation', 'Cardiometabolic_II'
runs[].units[].included	Whether the run unit should be imported as included	bool	no	Default is true for the first specified run unit for each plateLayout-panel pair, false otherwise

Example 1

```
explore-cli create -i /path/to/input/file.json -o path/to/projectfolder
```

2.5.3 Verb: export

Loads a project and exports the files specified.

Short option	Long option	Required	Description
-i	--input	yes	Path to project folder.
-o	--output	no	Name of folder in which to save files, defaults to working directory if not set.
	--npx	no	Exports NPX file.
	--extended-npx	no	Exports Extended NPX file.
	--cv-table	no	Exports the CV table file.
	--analysis-report	no	Exports analysis report on pdf format.
	--analysis-report-json	no	Exports analysis report on json format.

	--all	no	True if all available artifacts should be exported, if false only the specified files will be exported.
	--prefix	no	Sets the prefix of the output file names, default is the project name.
	--datetime-format	no	The format string for the datetime in exported file names (default = 'yyyy-MM-dd'), empty string results in no datetime being added to the filename.

This verb operates on an Explore project folder to retrieve relevant output files. More than one file at a time may be exported by including one or more argument switches. If no file switches are provided the program exports the default project data file in [*Apache Parquet*] format.

For documentation of NPX file and Extended NPX file, please refer to the user guide for *Olink® NPX Explore User Manual*.

Output files per default follow the naming convention: {PROJECT_NAME}_{FILE_TYPE}_{DATETIME}.{FILE_EXTENSION}

- The "PROJECT_NAME" portion may be altered with the `prefix` argument
- The "DATETIME" portion may be altered with the `datetime-format` argument

Example 1

```
explore-cli export -i path/to/projectfolder -o path/to/output --npx --analysis-report
```

2.5.4 Verbs: readme

Prints out the README for Olink Explore CLI in markdown format.

Short option	Long option	Required	Description
-v	--veb	no	The specific verb to display README text for, leave out to generate the entire README.

3. Appendix

3.1 Project data file (Apache Parquet)

The parquet file contains multiple columns and here follows a table explaining each column.

- The **Name** column presents the column names in the parquet file.
- The **Scope** column present which level the data occurs. For example if the scope is project, then that value applies to the whole project. If the scope value instead is data point then the value is unique for each data point.
- The **Type** column presents the data type of the values in the column.
- The **Example** column presents an example of what the data in the column could look like.
- The **Description** column provides a small description of the column.

Name	Scope	Type	Example	Comment
ProjectName	project	string	Project_1	Project name set by user
ProductType	project	string	Explore3072	Product type of project
Normalization	project	string	Plate control	Chosen by user. Can be Intensity or Plate control and affects reported NPX and LoD but not QC.
SampleMatrix	project	string	EDTA plasma	Sample matrix set by user
RunId	run unit	string	cad045cc-70ef-4479-8d1a-7b5973ab2aa7	Unique run unit identifier read from pre-processing run_metadata.json
ExperimentName	run unit	string	LJ111-1111_ SS123456_NEU_INF	NGS sequencing experiment name read from preprocessing run_metadata.json
FlowcellId	run unit	string	HHCYVDRXY	NGS run flowcell identifier read from pre-processing run_metadata.json
FlowcellType	run unit	string	S4	NGS run flowcell type read from pre-processing run_metadata.json
FlowcellSide	run unit	string	B	NGS run flowcell side read from pre-processing run_metadata.json
AnalyzerId	run unit	string	A01234	NGS instrument identifier read from pre-processing run_metadata.json. Displayed in Extended NPX as **AnalyzerID**.
InstrumentType	run unit	string	NovaSeq	NGS instrument type read from pre-processing run_metadata.json
InstrumentRunNumber	run unit	int	234	NGS instrument run number read from pre-processing run_metadata.json
SequencingStartTimestamp	run unit	DateTime	1999-12-31 23:00:00	Input read from preprocessing run_ metadata.json, UTC time. The date part is displayed in Extended NPX as **Processing_StartDate.**

Name	Scope	Type	Example	Comment
SequencingEndTimestamp	run unit	DateTime	1999-12-31 23:00:00	Input read from preprocessing run_metadata.json, UTC time. The date part is displayed in Extended NPX as **Processing_EndDate**.
PreProcessingRunTimestamp	run unit	DateTime	2001-01-01 02:00:00	Input read from preprocessing run_metadata.json, UTC time
PreProcessingVersion	run unit	string	2.2.0	Version of pre-processing software, read from run_metadata.json
SequencingRecipeName	run unit	string	Olink_NovaSeq6K_S4_V1	NGS instrument setting read from pre-processing run_metadata.json
LibraryNumber	run unit	int	1	NGS library which maps to lane on S4 flowcell
IndexPlate	run unit	string	1	Sample index plate, i.e. range of sample indices, for run unit. Can be '1' or '2 or '3' or '4' in Explore 3072.
SampleIndexVersion	run unit	int	1	Sample index version used in counts file. Version is 1 for all Olink Explore 3072 index plates except Olink Explore Index plate with part number 87005 and EXPD 2023-05-31 which is version 0.
EstimatedPanel	run unit	string	Neurology	Automatically detected panel by preprocessing software, read from run_metadata.json
MatchedCounts	library	long	354689590	Input read from pre-processing run_metadata.json
Reads	library	long	638337024	Input read from pre-processing run_metadata.json
ReadsPf	library	long	503632672	Input read from pre-processing run_metadata.json
PercentReadsPf	library	double	78.89762115478516	Input read from pre-processing run_metadata.json
RunUnitId	run unit	string	8ca76722-d1fd-4a4a-a296-d77415675651	Unique run unit identifier read from pre-processing run_metadata.json
PanelLotNr	run unit	string	B04406	Panel lot number entered by user, deciding which reference values to use for Quality control metrics. Displayed in NPX file and Extended NPX as **Panel_Lot_Nr**
Panel	run unit	string	Neurology	Panel of chosen PanelLotNr
PlateId	plate	string	SS123456	Name of plate layout file without extension. Displayed in NPX file and Extended NPX as **PlateID**

Name	Scope	Type	Example	Comment
Included	run unit	bool	TRUE	Default true but can be set to false and then run unit is not included in any calculations
ExploreVersion	run unit	string	1.6.0.0	Version of Olink Explore calculation module
WellIndex	sample	Int	1	Integer identifier of well on 96-well plate with row-wise numbering. Displayed in NPX file and Extended NPX as **Index**.
WellId	sample	string	A1	Well on 96-plate as given in plate layout file. Displayed in Extended NPX as **WellID.**
SampleId	sample	string	subject-123	Sample identifier which must be unique within Explore project. Displayed in NPX file and Extended NPX as **SampleID**.
SampleType	sample	string	SAMPLE	Sample type as given in plate layout file.
Block	block	string	B	Dilution block of assay.
BlockQcFlag	block-plate	int	0	A numeric representation of the QC flags displayed on Quality control tab in MyData and NPX Explore. BlockQcFlag is computed as $numSamplesWarnFlag x 4$ + $madPlateCtrlFlag x 16$ + $medianZScoreNegCtrlFlag x 32$ + $medianZScorePlateCtrlFlag x 64$ where each individual flag is set to 1 if the corresponding criteria flags, and 0 otherwise.
NumSamplesWarn	block-plate	int	5	The sum of low counts sample warnings and manual QC warnings for the block and plate. Displayed in view 'Quality control' in Olink MyData and Olink NPX Explore
MedianZScoreNegCtrl	block-plate	double	2.122345	The numeric value of Median Z-score Negative control. Displayed in view 'Quality control' in Olink MyData and Olink NPX Explore
MedianZScorePlateCtrl	block-plate	double	-0.17504	The numeric value of Median Z-score Plate control. Displayed in view 'Quality control' in Olink MyData and Olink NPX Explore
MadPlateCtrl	block-plate	double	0.52877	The numeric value of MAD Zscore Plate control. Displayed in view 'Quality control' in Olink MyData and Olink NPX Explore

Name	Scope	Type	Example	Comment
SampleBlockQcWarn	block-sample	int	0	An integer representation of the sample-block QC warnings presented as text in NPX file and Extended NPX file in column **QC_Warning.** Possible values are 0: PASS, 8: WARN (due to low counts), 16: MANUAL_WARN (set via Olink MyData or Olink NPX Explore).
OlinkId	assay	string	OID20790	Olink assay identifier.
Assay	assay	string	SESTD1	Assay name displayed in Olink MyData and Olink NPX Explore.
AssayCategory	assay	int	0	0: regular assay with results exported in NPX file and Extended NPX file 1: assay did not meet Olink's batch release quality control criteria in current lot and is therefore excluded from analysis and listed in NPX file as EXCLUDED 2, 3, 4: data point does not represent assay that is part of Olink Explore product.
Uniprot	assay	string	Q86VW0	Uniprot ID.
NegCtrlZScore	assay-plate	double	2.253177	Z-score value for negative control. Displayed in view 'Control strip' and is input to median calculation for the Quality control' view in Olink MyData and Olink NPX Explore
PlateCtrlZScore	assay-plate	double	-0.91364	Z-score value for plate control. Displayed in view 'Control strip' and is input to median calculation for the 'Quality control' view in Olink MyData and Olink NPX Explore
LodPcNorm	assay-plate	double	2.074716	Limit of detection for assay on plate displayed in column **LOD** of NPX file and Extended NPX file if plate control normalization has been chosen.
LodIntNorm	assay-plate	double	0.850304	Limit of detection for assay on plate displayed in column **LOD** of NPX file and Extended NPX file if intensity normalization has been chosen.
IntraCvPcNorm	assay-plate	double	0.101	Intra-plate CV for assay based on CONTROL samples when plate control normalization has been chosen Displayed in Extended NPX file in column **IntraCV** if normalization is Plate control.

Name	Scope	Type	Example	Comment
IntraCvIntNorm	assay-plate	double	0.101	Intra-plate CV for assay based on CONTROL samples when intensity normalization has been chosen Displayed in Extended NPX file in column **IntraCV** if normalization is Intensity.
InterCvPcNorm	assay-project	double	0.201	Project CV for assay based on CONTROL samples when plate control normalization has been chosen. Displayed in Extended NPX file in column **InterCV** if normalization is Plate control.
InterCvIntNorm	assay-project	double	0.201	Project CV for assay based on CONTROL samples when intensity normalization has been chosen. Displayed in Extended NPX file in column **InterCV** if normalization is Intensity.
MissingFreq	assay-project	double	0.943182	Missing frequency presented in NPX and Extended NPX file in column **MissingFreq.**
AssayQcWarn	assay-plate	int	0	An integer representation of the assay warning presented in NPX and Extended NPX file in column **Assay_Warning**. Possible values are 0: PASS, 1: WARN.
NPXPcNorm	data point	double	1.735509	NPX value displayed in **NPX** column of NPX file and Extended NPX file if plate control normalization has been chosen.
NPXIntNorm	data point	double	0.511097	NPX value displayed in **NPX** column of NPX file and Extended NPX file if intensity normalization has been chosen. NB: assays with bimodal distribution will always be plate control normalized.
ExtNPX	data point	double	-1.94701	Intermediate value between count and NPX: log2 of the ratio between data point Count value and the count for the Extension Control assay for the same sample.
Counts	data point	int	2641	Counts from pre-processing generated counts file, data points for category 1 assays (excluded assays) are reported as 0.

4. Revision history

Version	Date	Description
1.3.0	2024-03-07	*1.2*, *2*, and *2.1* requirement updated to .Net 8 runtime.
1.1	2023-06-19	Appendix table updated for Counts
1.0	2023-04-12	MasterPlate removed
1.0 Beta	2023-03-14	New

www.olink.com

Olink products and services are For Research Use Only and not for Use in Diagnostic Procedures.
All information in this document is subject to change without notice. This document is not intended to convey any warranties, representations and/or recommendations of any kind, unless such warranties, representations and/or recommendations are explicitly stated.
Olink assumes no liability arising from a prospective reader's actions based on this document.
OLINK, NPX, PEA, PROXIMITY EXTENSION, INSIGHT and the Olink logotype are trademarks registered, or pending registration, by Olink Proteomics AB. All third-party trademarks are the property of their respective owners.
Olink products and assay methods are covered by several patents and patent applications *https://www.olink.com/patents/*