Olink®

NPX™ Explore CLI HT & 3072

Technical Information

1399, 2.3.3, 2024-03-13

Table of contents

1. Introduction 3
   - 1.1 Olink® Explore Software terminology and documentation 3
   - 1.2 Requirements 4
2. Installation 5
   - 2.1 Self-contained executable 5
   - 2.2 Docker or podman 5
   - 2.3 Logging 6
   - 2.4 Explore projects 6
   - 2.5 Verbs 8
3. Appendix 14
   - 3.1 Project data file (Apache Parquet) 14
4. Revision history 17

1. Introduction

NPX™ Explore CLI HT & 3072 is a command-line interface (cli) for the *Olink® Explore HT* and the *Olink® Explore 3072* 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 Explore software terminology and output files, please refer to the user guide for [*NPX™ Explore HT & 3072 software*, which shares the same Explore software library for normalization, QC and output file generation.

1.1.1 NGS run folder

NGS sequencing output from a single flowcell.

1.1.2 Pre-processing

Conversion of NGS output to counts per Olink Explore index-barcode sequence.

1.1.3 Plate layout file

A csv-file containing sample ID and sample type per well of one 96-well plate.

1.1.4 Run unit

A set of one plate layout and a Block, the smallest set that can be analyzed in the Explore software. One counts file from the pre-processing contains counts for one run unit: i.e. one 96-well-plate and one block.

1.1.5 Project

A set of run units and all associated sample, block and QC data, plus project metadata. Projects are, by design, independent of each other in Explore software. Project summary statistics (Inter-CV, CV distribution per Block) depend on included run units.

Output file	Description
Analysis Report	A pdf-file with a summary of project metrics.
NPX™ file	A parquet-file with one record per combination of sample/control/ external control and assay/internal control.
Extended NPX™ file	A parquet-file containing the same columns as the NPX file with additional columns.
CLI Data Export file	A parquet-file containing the same columns as the extended NPX file with additional columns and records for empty wells and excluded run units.

1.2 Requirements

The program has the following system requirements:

- A *Net 8 runtime* installed on the system (unless using the self-contained version or the docker image).
- 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 or podman

The CLI is available as a Docker image for environments demanding it. Usage of Explore CLI through Docker or podman 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:

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

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

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

```
docker run --rm localhost/explore-cli:<VERSION> info
```

For podman run the following command:

```
podman run --rm localhost/explore-cli:<VERSION> info
```

where VERSION is the downloaded version of the explore-cli such as `2.1.0`.

2.2.2 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. For podman change `docker` to `podman`.

```
docker run --rm \
--mount type=bind,source="${PWD}",target=/data \
-u $(id -u ${USER}):$(id -g ${USER}) \
localhost/explore-cli:<VERSION> 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 (NPX™ Explore HT & 3072) 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 include:
  - 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.oep
```

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.oep	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 blocks).

2.5 Verbs

The Olink Explore 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.normalizationAndQcSpecification	The version of the specification for normalization and QC calculations.
versions.outputFileFormat	The version of the data output formats
dataAnalysisReferenceIds.ExploreHT	The list of available data analysis reference IDs in this version of the software for Explore HT
dataAnalysisReferenceIds.Explore3072	The list of available data analysis reference IDs in this version of the software for Explore3072

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	A
097	192	B
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 blocks used in the project must be mapped to one data analysis reference id in the dictionary `selectedDataAnalysisRefIds`, and all run units in the project must be mapped to exactly one block. Run units with block not included in `selectedDataAnalysisRefIds` will cause an error and abort the operation.

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

Example json input format (contract version 2)

```
{
    "Version": 2,
    "projectName": "TestProject-CSAS3_EXPL",
    "productType": "ExploreHT",
    "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"
    },
    "selectedDataAnalysisRefIds": {
          "Block_1": "D10001",
          "Block_2": "D20001",
          "Block_3": "D30001",
          "Block_4": "D40001",
          "Block_5": "D50001",
          "Block_6": "D60001",
          "Block_7": "D70001",
          "Block_8": "D80001"
    },
    "plateLayouts": [
          {
                "path": "plate_layouts/plate1.csv",
          }
    ],
    "runs": [
          {
                "path": "230405_A00915_0850_AHYMWNDSX3",
                "units": [
                      {
                            "plateLayout": "plate1",
                            "libraryNumber": 1,
                            "indexPlate": "A",
                            "block": "Block_1",
                            "included": true
                      }
                      {
                            "plateLayout": "plate1",
                            "libraryNumber": 1,
                            "indexPlate": "A",
                            "block": "Block_2",
```

```
                                "included": true
                            }
                        ]
                    }
                ]
            }
```

Field descriptions

Field	Description	Data type	Required	Comment
version	The version of the json input file contract	int	yes	Supported versions: 2
projectName	Name of the project	string	yes	
productType	Type of product used in project	string	no	'ExploreHT' (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	
selectedDataAnalysis RefIds	Dictionary of block to data analysis reference id selections	Dictionary<string,string>	yes	ex. 'Block_1': 'D10001', 'Cardiometabolic': 'B14808'
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	string	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	string	yes	Folder or zip file 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	string	yes	Must be the plateId of a plate layout defined under the plate.
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	string	yes	'A, 'B', '1', '2', '3', '4'
runs[].units[].block	Supposed panel of the run unit	string	yes*	ex. 'Block_1', 'Block_8'. *Required if productType is ExploreHT
runs[].units[].panel	Supposed panel of the run unit	string	no*	ex. 'Cardiometabolic', 'Inflammation_II'. *Only applicable if productType is Explore3072
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. Defaults to exporting a CLI Data Export file.

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.
	--analysis-report	no	Exports analysis report on pdf format.
	--analysis-report-json	no	Exports analysis report on json format.
	--npx	no	Exports the NPX file on parquet format.
	--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.
	--extended-npx	no	Exports the Extended NPX file on parquet format.
	--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.

	--warnings-as-errors	no	True if warnings in the project should fail the operation, otherwise they will only be logged.

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 HT & 3072 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 NPX™ Explore CLI HT & 3072 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 datapoint then the value is unique for each datapoint.
- 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
SampleID	sample	string	subject-123	Sample identifier which must be unique within Explore project.
SampleType	sample	string		Sample type as given in plate layout file.
WellID	sample	string	A1	Well on 96-plate as given in plate layout file.
PlateID	plate	string	SS123456	Name of plate layout file without extension.
DataAnalysisRefID	run unit	string	D10001	Data analysis reference id entered by user, deciding which reference values to use for Quality control metrics.
OlinkID	assay	string	OID20790	NGS sequencing experiment name read from preprocessing run_metadata.json.
UniProt	assay	string	Q86VW0	Uniprot ID
Assay	assay	string	SESTD1	Assay name displayed in NPX Explore.
AssayType	assay	string	assay	Assay type of datapoint. Possible values are **assay, amp_ctrl, inc_ctrl, ext_ctrl**.
Panel	run unit	string	Explore_HT for Explore HT and Cardimetabolic etc. for Explore3072	Explore_HT
Block	Block	string	1	Dilution block
Count	data-point	int	2641	Counts from pre-processing generated counts file.
ExtNPX	data-point	double	-1.94701	Intermediate value between count and NPX: log2 of the ratio between datapoint Count value and the count for the Extension Control assay for the same sample.
NPX	data-point	double	1.735509	NPX for project chosen normalization. Bimodal assays are always plate control normalized.

Name	Scope	Type	Example	Comment
PCNormalizedNPX	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.
AssayQC	data-point	string	PASS	String representation of the AssayQCWarn column.
SampleQC	data-point	string	PASS	Combined string representation of the SampleBlockQcStatus, SampleBlockQCFail and BlockQCFail columns.
ExploreVersion	run unit	string	5.0.0	NGS library which maps to lane on S4 flowcell.
IntraCV	assay-plate	double	0.101	Intra-plate CV for assay based on CONTROL samples when plate control normalization has been chosen.
InterCV	assay-project	double	0.201	Project CV for assay based on CONTROL samples when plate control normalization has been chosen.
SampleBlockQCWarn	data-point	int	0	Sample-Block QC warn for this datapoint.
SampleBlockQCFail	data-point	int	0	Sample-Block QC fail for this datapoint.
BlockQCFail	data-point	int	0	Block QC fail for this datapoint.
ReadsPf	data-point	int	0	Assay QC warning for each assay.
RunID	run unit	string	8ca76722-d1fd-4a4a-a296-d77415675651	Unique identifier of run 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.
ExperimentName	run unit	string	LJ111-1111_ SS123456_NEU_INF	NGS sequencing experiment name read from pre-processing 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.
InstrumentID	run unit	string	A01234	NGS instrument identifier read from pre-processing run_metadata.json.
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 pre-processing run_metadata.json, UTC time.

Name	Scope	Type	Example	Comment
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	A	Sample index plate, i.e. range of sample indices, for run unit. Can be **A** or **B** in Explore HT and '1', '2', '3', '4' for Explore 3072.
SampleIndexVersion	run unit	int	2	Sample index version used in counts file. Version is 2 for all Explore HT index plates and 1 for Explore3072 index plates.
MatchedCounts	library	long	354689590	Input read from pre-processing run_metadata.json.
Reads	library	long	638337024	Input read from pre-processing run_metadata.json.
Included	run unit	bool	TRUE	Default true but can be set to false and then run unit is not included in any calculations.
PreProcessingRunTimestamp	run unit	DateTime	2001-01-01 02:00:00	Input read from pre-processing run_metadata.json, UTC time.
PreProcessingVersion	run unit	string	4.0.0	Version of pre-processing software, read from run_metadata.json.
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 for current data analysis reference ID and is therefore excluded from analysis and listed in NPX file as EXCLUDED.
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

4. Revision history

Version	Date	Description
2.3.3	2024-03-12	Updated name of softwares to NPX™ Explore HT & 3072 and NPX™ CLI HT & 3072. *1.2*, *2,* and *2.1* updated to NET 8 runtime. *2.5.1* Explore 3072 added. *2.5.2* second and third table updated. *3* table updated.
2.2.0	2023-12-01	*2.5.1* table updated.
2.1.0	2023-10-09	*2.2* updated with podman. *2.5.2* runs[].path updated. *2.5.3* --warnings-as-errors added.
2.0.0	2023-08-14	New

www.olink.com

1399, 2.3.3, 2024-03-13