

Implementation of Python Large Optic Stitching (PyLOSt) software with orange data mining platform using Python 3.8+

F. Perrin, ESRF

Contents

1	Initial Installation	1
1.1	Basic user instructions:.....	2
1.2	Advanced user instructions:	2
1.2.1	Alternative 1 - Orange3 standalone installer:	2
1.2.2	Alternative 2 – Installation of Orange3 from within conda environment:.....	3
1.2.3	Final steps - PyLOSt installation in the Orange3 environment:.....	4
2	Updates and developer mode instructions:	6
2.1	pip install + git	6
2.2	git clone + develop	7
3	Bugs & issues:	9

1 Initial Installation

In order to have a functional PyLOSt toolbox for Orange data mining some requirements must be met:

- An Orange3 version above 3.31.0 (either installed from the standalone installer (<https://orangedatamining.com/download>) or through a pip/conda install command)
- Some additional packages have to be present where Orange is installed:
 - o conda
 - o *setuptools* (should come with any python distribution 3.8 and above)
 - o sqlalchemy
 - o scikit-image
 - o line_profiler
 - o **numpy < 1.23 (important until further updates of PyLOSt code)**
 - o h5py & hdf5 (be aware that mismatch between MPI version & no MPI will lead to Pylost tools failing to appear)
 - o pyFAI
 - o sympy
 - o **pytorch** (conda install) or **torch** (pip install)
 - o astropy
 - o silx

The order of the list above can be used to build a separated python environment dedicated to PyLOSt (*see advanced user instructions*).

Once these packages are installed one can install the two specific pylost packages pylost (which contains all of the data processing) and pylost-orange (which configures the interactions/widgets between pylost and the orange interface).

To install the necessary software (Orange and PyLOSt) there are 2 basic approaches:

1. use a standalone installer (basic user instructions)
2. install and configure the packages 'manually' (advanced user instructions)

1.1 Basic user instructions:

Regular installers for Windows 64bits and Linux 64bits have been created to allow quick and reliable installations. These installers will create specific 'conda' environments which should then be protected from overwriting by subsequent software installations on the PC. They can be download from the DESY cloud:

LEAPS WG1.2>MoonPics Metrology>PyLOSt.

The win-64 installer comes in two variants:

- A small size executable (~60MB) which creates an independent minimal python environment and then retrieves the packages required from public external conda repositories (*warning: future dependencies between packages could be broken as they are updated*). **This will only work with an internet connection.**
- A full installer (~900MB) which allow an installation offline. The environment is stable and works out of the box.

The linux-64 installation script only comes in a ~800MB file and is similar to the full win-64 installer.

PyLOSt updates are not easily implemented using this installation method. Currently if after initial installation it is needed to update the installation, you will need to refer to the advanced installation methods described below.

1.2 Advanced user instructions:

1.2.1 Alternative 1 - Orange3 standalone installer:

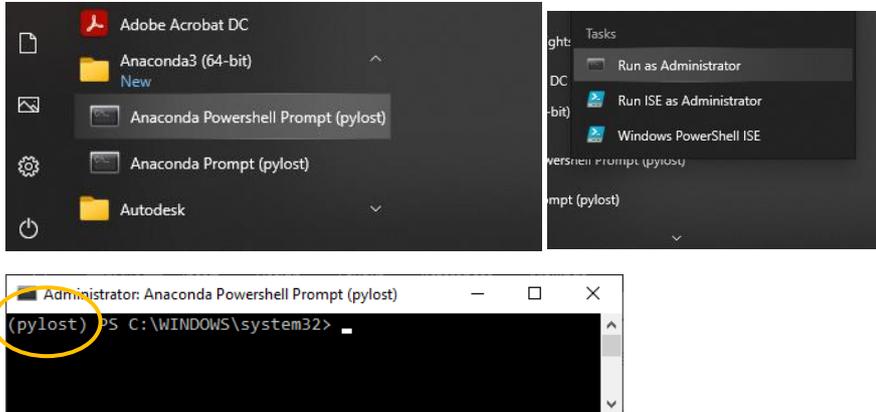
This method can be used following a new fresh install of Orange data mining from their installer (<https://orangedatamining.com/download>) or on an existing installation of Orange above 3.30 if dependencies are carefully set.

The major drawback is that some dependencies for PyLOSt may not be correct as the installer comes with its own packages and some will be need to be replaced with PyLOSt compatible versions afterward. Some issues have also been seen after updating Orange3 from within the GUI. Consequently the following alternative is recommended since the orange3 package is installed at the very end and will not interfere with pyFAI, silx, pytorch, etc. where dependencies can conflict.

With Orange installed, the final installation steps (described in "[Final steps - PyLOSt installation in the Orange3 environment](#)") are executed from the Orange command prompt in the start menu.

Command prompt:

The following commands should be entered via the specific command prompt, which is located in the start menu. Be sure to use “Run as Administrator” if needed.



The current target environment can be checked by verifying that its name is inside parenthesis. No parenthesis means that conda command is not recognized by the operating system (see [Bugs & issues – PATH variable](#)).

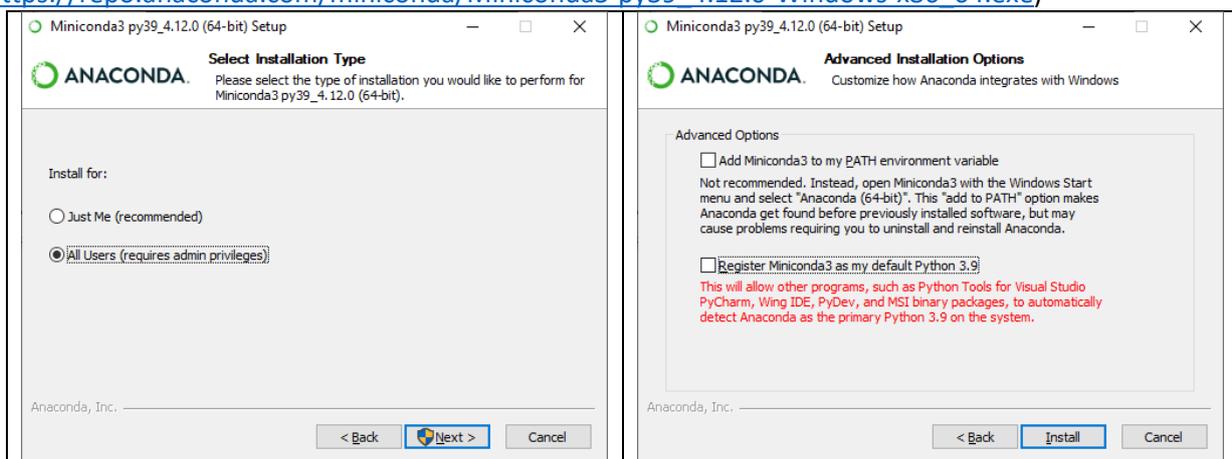
1.2.2 Alternative 2 – Installation of Orange3 from within conda environment:

The use of conda environments (<https://conda.io/projects/conda/en/latest/user-guide/concepts/environments.html>) is highly recommended as they encapsulate all the requirements without interfering with other python-based applications and are more resilient to dependency checks.

Alternative methods may not be supported (pip, anaconda GUI, mamba, etc.).

Download & install the miniconda installer

(https://repo.anaconda.com/miniconda/Miniconda3-py39_4.12.0-Windows-x86_64.exe)



Open (with administrator rights if needed as above) the terminal activating the base conda environment using the link to the command prompt created in the start menu and from here create

the first batch of a PyLOSt/Orange3 dedicated environment (named *pylost* in this manual but it can be any prefix):

```
Administrator: Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\WINDOWS\system32> conda create --name pylost python=3.9 conda setuptools console_shortcut powershell_shortcut
```

`conda create --name pylost python=3.9 conda setuptools console_shortcut powershell_shortcut`

The next step consists of adding all the remaining packages needed for PyLOSt to this new environment. By using either the command line **conda activate pylost** or using the newly created link to the command prompt in the start menu:

```
Administrator: Anaconda Powershell Prompt (Miniconda3)
conda-4.14.0 | 937 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
# $ conda activate pylost
#
# To deactivate an active environment, use
#
# $ conda deactivate
#
(base) PS C:\WINDOWS\system32> conda activate pylost
(pylost) PS C:\WINDOWS\system32>
```

`conda activate pylost` will switch, in the same command prompt, to the *pylost* environment just created from the base environment created by the miniconda installer.

1.2.3 Final steps - PyLOSt installation in the Orange3 environment:

This method is common to both alternatives.

1.2.3.1 Installation of the remaining packages

```
Administrator: Anaconda Powershell Prompt (pylost)
(pylost) PS C:\WINDOWS\system32> conda install -c conda-forge sqlalchemy scikit-image line_profiler "numpy<1.23" h5py hdf5 hdf5plugin pyFAI sympy pytorch astropy silx orange3
```

`conda install -c conda-forge sqlalchemy scikit-image line_profiler "numpy<1.23" h5py hdf5 hdf5plugin pyFAI sympy pytorch astropy silx orange3`

Notice the order and some version specification. Here `numpy` will be forced to version before 1.23 and the use of the channel **conda-forge** will set the `h5py` and `hdf5` not using MPI ([Message Passing Interface](#)) not fully asserted for PyLOSt at this moment.

```
Administrator: Anaconda Powershell Prompt (pylost)
gstreamer conda-forge/win-64::gstreamer-1.18.5-hdff456e_3
gts conda-forge/win-64::gts-0.7.6-h7c369d9_2
h11 conda-forge/noarch::h11-0.12.0-pyhd8ed1ab_0
h2 conda-forge/win-64::h2-4.1.0-py39hc45309_0
h5py conda-forge/win-64::h5py-3.7.0-nompi_py39h7fe8f52_101
harfbuzz pkgs/main/win-64::harfbuzz-4.3.0-hda2c7e1_0
hdf5 conda-forge/win-64::hdf5-1.12.2-nompi_h2a0e4a3_100
hdf5plugin conda-forge/win-64::hdf5plugin-3.2.0-py39hf79b81a_1
hpack conda-forge/noarch::hpack-4.0.0-pyh0f0ad1d_0
httpcore conda-forge/noarch::httpcore-0.15.0-pyhd8ed1ab_0
httpx conda-forge/win-64::httpx-0.23.0-py39hcbf5309_1
hyperframe conda-forge/noarch::hyperframe-6.0.1-pyhd8ed1ab_0
icu anaconda/win-64::icu-58.2-vc14hc45fdbb_0
```

As previously indicated `h5py` and `hdf5` packages must come from the same channel and some precautions must be taken when installing new packages or updating the environment.

1.2.3.2 Installation of PyLOSt source code

PyLOSt code is currently only accessible from the ESRF gitlab as a public project (no credentials needed): https://gitlab.esrf.fr/moonpics_stitching_2018

git protocol must be installed within the operating system: (<https://git-scm.com/downloads>).

Subsequently, from the command prompt of the target environment (orange or pylost) **pip** and **git** can easily download and install the source code:

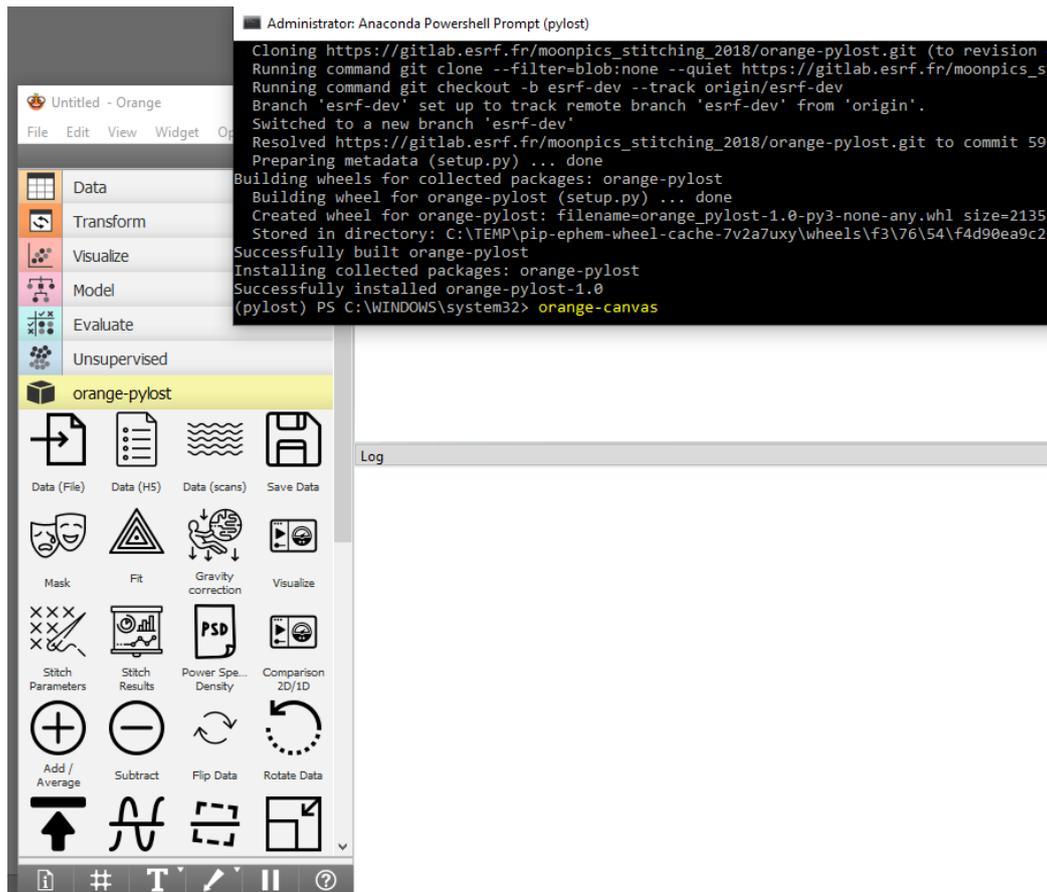
```
Administrator: Anaconda Powershell Prompt (pylost)
Collecting git+https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git@esrf-dev
Cloning https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git (to revision esrf-dev) to c:\temp\pip-req-build-0ky4cxtd
Running command git clone --filter=blob:none --quiet https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git 'C:\TEMP\pip-req-build-0ky4cxtd'
Running command git checkout -b esrf-dev --track origin/esrf-dev
Branch 'esrf-dev' set up to track remote branch 'esrf-dev' from 'origin'.
Switched to a new branch 'esrf-dev'
Resolved https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git to commit cd9ec7b7e9222c15c102eb1a9e4f6fffeaba6d557
Preparing metadata (setup.py) ... done
(pylost) PS C:\WINDOWS\system32> pip install --no-deps git+https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git@esrf-dev
Collecting git+https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git@esrf-dev
Cloning https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git (to revision esrf-dev) to c:\temp\pip-req-build-xner6kpp
Running command git clone --filter=blob:none --quiet https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git 'C:\TEMP\pip-req-build-xner6kpp'
Running command git checkout -b esrf-dev --track origin/esrf-dev
Branch 'esrf-dev' set up to track remote branch 'esrf-dev' from 'origin'.
Switched to a new branch 'esrf-dev'
Resolved https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git to commit 59af79ee90644e37b56e50601b87d9edabe2f46
Preparing metadata (setup.py) ... done
Building wheels for collected packages: orange-pylost
Building wheel for orange-pylost (setup.py) ... done
Created wheel for orange-pylost: filename=orange_pylost-1.0-py3-none-any.whl size=213531208 sha256=3595a81c90b6f09d35c0cd63f8658768a6986eabf4dc563f7b20cc53870cf5486
Stored in directory: C:\TEMP\pip-ephem-wheel-cache-7v2a7uxy\wheels\f3\76\54\4d90ea9c2b2d11e0645a2e19a1a6a4c462796a001648f78a7
Successfully built orange-pylost
Installing collected packages: orange-pylost
Successfully installed orange-pylost-1.0
(pylost) PS C:\WINDOWS\system32>
```

```
pip install --no-deps git+https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git@master
pip install --no-deps git+https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git@master
```

*the option **--no-deps** is important to avoid some issues with the dependencies check.*

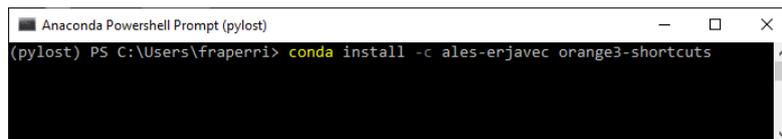
*...**@master** force the installation of the master branch, another branches are available (esrf-dev, etc.) and can be download using this syntax.*

Now you should have working PyLOSt tools for Orange Data Mining!!!



orange-canvas to open a new Orange session from the command prompt.

In addition, an Orange3 desktop shortcut can be created anytime automatically:



`conda install -c ales-erjavec orange3-shortcuts`

2 Updates and developer mode instructions:

2.1 pip install + git

Both previous methods install a version of PyLOSt, which cannot be updated as easily as well-known python packages through repository managers. An advanced package management will be maybe available in the future but for now, the simplest way is to reinstall both **PyLOSt** and **orange-pylost** in the pylost environment (if you are in the correct environment console/powershell prompt and its conda command recognized you should see *(environment prefix)* at the start of the command line):

```

Administrator: Anaconda Powershell Prompt (pylost)
(pylost) PS C:\WINDOWS\system32> pip uninstall PyLOSt
Found existing installation: PyLOSt 0.3
Uninstalling PyLOSt-0.3:
  Would remove:
    c:\programdata\miniconda3\envs\pylost\lib\site-packages\pylost-0.3.dist-info*
    c:\programdata\miniconda3\envs\pylost\lib\site-packages\pylost*
Proceed (Y/n)? y
Successfully uninstalled PyLOSt-0.3
(pylost) PS C:\WINDOWS\system32> pip install --no-deps git+https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git@master
Collecting git+https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git@master
  Cloning https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git (to revision master) to c:\temp\pip-req-build-htqa_30p
  Running command git clone --filter=blob:none --quiet https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git 'C:\TEMP\pip-req-build-htqa_30p'
  Resolved https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git to commit 54ab8b2358a93cc9c99ea2ba958d376fafa97346
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: PyLOSt
  Building wheel for PyLOSt (setup.py) ... done
  Created wheel for PyLOSt: filename=PyLOSt-0.3-py3-none-any.whl size=248442 sha256=9700613a128e178c57d2aa00de5371fde126a8fd7c0b90fd2bccf28ba5839d26
  Stored in directory: C:\TEMP\pip-ephem-wheel-cache-ik6ncuhj\wheels\f4\c7\48\becfefcf1a2b486ff3b3bcf9a5accf459c74c0818d7d47fe2f
Successfully built PyLOSt
Installing collected packages: PyLOSt
Successfully installed PyLOSt-0.3
(pylost) PS C:\WINDOWS\system32> pip uninstall orange-pylost
Found existing installation: orange-pylost 1.0
Uninstalling orange-pylost-1.0:
  Would remove:

```

```

pip uninstall PyLOSt
pip install --no-deps git+https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git@master
pip uninstall orange-pylost
pip install --no-deps git+https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git@master

```

2.2 git clone + develop

Another way is to install PyLOSt and orange-pylost in develop mode. Unlike the **pip + git install**, which only download the source code, **git clone** will also import the tracking of all the changes made on the source code. A simple **git pull --all -v** command can be then used to get the latest modifications made on the source code by maintainers. In addition, rolling back to a previous version is also possible (not explained in this document).

Any installation of PyLOSt described above, even by the installers, can be switched to a developer mode by uninstalling the PyLOSt and orange-pylost packages prior to the following sequence of commands:

- Create a directory somewhere for the PyLOSt package:
`mkdir c:\python_test\pylost`
- Clone sources with git, branch can be set here also:
`git clone https://gitlab.esrf.fr/moonpics_stitching_2018/PyLOSt.git --branch esrf-dev c:\python_test\pylost`
- Fetch from the directory to check the existence of new modifications on the source code:
`cd C:\python_test\pylost`
`git fetch --all -v`
- or Pull from the directory (fetch is implicit) to get the latest changes:
`cd C:\python_test\pylost`
`git pull --all -v`

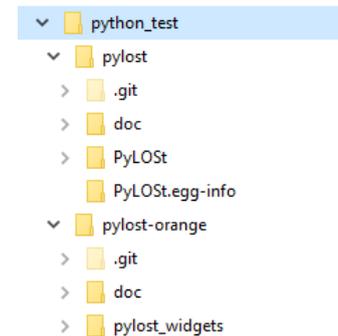
```

Administrator: Anaconda Powershell Prompt (pylost)
(pylost) PS C:\python_test\pylost> git clone https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git --branch esrf-dev c:\python_test\pylost-orange
Cloning into 'c:\python_test\pylost-orange'...
remote: Enumerating objects: 3739, done.
remote: Counting objects: 100% (629/629), done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 3739 (delta 461), reused 608 (delta 445), pack-reused 3110
Receiving objects: 100% (3739/3739), 208.94 MiB | 11.26 MiB/s, done.
Resolving deltas: 100% (2824/2824), done.
Updating files: 100% (361/361), done.
(pylost) PS C:\python_test\pylost> cd C:\python_test\pylost-orange
(pylost) PS C:\python_test\pylost-orange> git pull --all -v
Fetching origin
POST git-upload-pack (165 bytes)
POST git-upload-pack (968 bytes)
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 4), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 427 bytes | 1024 bytes/s, done.
From https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost
  59af79e..781e6b5  esrf-dev  -> origin/esrf-dev
   = [up to date]    master    -> origin/master
Updating 59af79e..781e6b5
Fast-forward
 pylost_widgets/learning/net/_init__.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 pylost_widgets/learning/net/_init__.py
(pylost) PS C:\python_test\pylost-orange>

```

The same must be done for the orange-pylost package in a different directory:

- Create a directory somewhere for the orange-pylost package:
`mkdir c:\python_test\pylost-orange`
- Clone sources with git, branch can be set here also:
`git clone https://gitlab.esrf.fr/moonpics_stitching_2018/orange-pylost.git --branch esrf-dev c:\python_test\pylost-orange`
- Fetch from the directory if needed:
`cd C:\python_test\pylost-orange`
`git fetch --all -v`
- or Pull from the directory (fetch is implicit):
`cd C:\python_test\pylost-orange`
`git pull --all -v`



The source code is now in the target directories and Python just needs to know where to look:

```

Administrator: Anaconda Powershell Prompt (pylost)
Creating c:\programdata\miniconda3\envs\pylost\lib\site-packages\PyLOSt.egg-link (link to .)
Adding PyLOSt 0.3 to easy-install.pth file

Installed c:\python_test\pylost
(pylost) PS C:\python_test\pylost> cd C:\python_test\pylost-orange
(pylost) PS C:\python_test\pylost-orange> python setup.py develop --no-deps
running develop
C:\ProgramData\Miniconda3\envs\pylost\lib\site-packages\setuptools\command\easy_install.py:144: EasyInstallDeprecationWarning:
  warning: easy_install command is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
C:\ProgramData\Miniconda3\envs\pylost\lib\site-packages\setuptools\command\install.py:34: SetuptoolsDeprecationWarning:
  warning: setup.py install is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
running egg_info
creating orange_pylost.egg-info
writing orange_pylost.egg-info\PKG-INFO
writing dependency_links to orange_pylost.egg-info\dependency_links.txt
writing entry_points to orange_pylost.egg-info\entry_points.txt
writing requirements to orange_pylost.egg-info\requires.txt
writing top-level names to orange_pylost.egg-info\top_level.txt
writing manifest file 'orange_pylost.egg-info\SOURCES.txt'
reading manifest file 'orange_pylost.egg-info\SOURCES.txt'
adding license file 'LICENSE'
writing manifest file 'orange_pylost.egg-info\SOURCES.txt'
running build_ext
Creating c:\programdata\miniconda3\envs\pylost\lib\site-packages\orange-pylost.egg-link (link to .)
Adding orange-pylost 1.0 to easy-install.pth file

Installed c:\python_test\pylost-orange
(pylost) PS C:\python_test\pylost-orange>

```

```
cd C:\python_test\pylost
python setup.py develop --no-deps
cd C:\python_test\pylost-orange
python setup.py develop --no-deps
```

Updating is performed by a simple pull from the ESRF gitlab server and changes will be active as soon as Orange is restarted.

This code can be modified locally and those modifications will be checked against the gitlab source code when pulling, **git** raising a warning if any conflict is detected.

3 Bugs & issues:

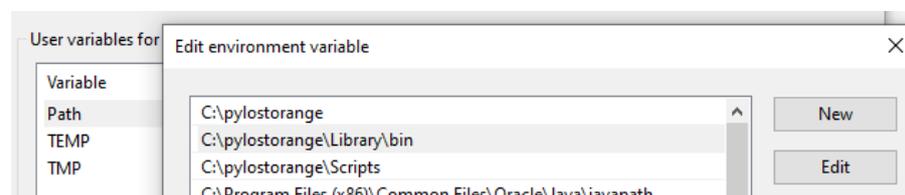
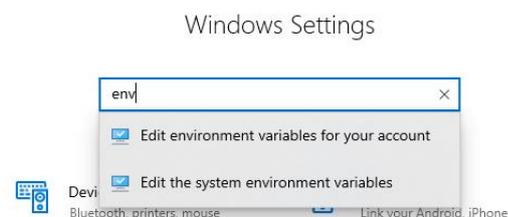
For some institutes, the proxy policies could prevent the access to the anaconda repositories, one can use the offline installers

If Orange can be launched but the PyLOSt tools do not appear, several causes have been spotted:

- o the PATH environment variable is not properly set. It should include at least those lines:

```
C:\...\[environment path]
C:\...\[environment path]\Scripts
```

On Windows settings, “edit environment variables for your account”



- o mismatch between hf5py & hfd5, or other
- o dependencies failure, missing packages (updates, etc.)
- o other...

An Orange log file can be found in the %USERPROFILE%\AppData\Local\Orange folder on Windows and ~/.local/share/Orange/ on Linux.

PyLOSt tools have been mostly developed and tested under Windows 10 system; some issues may appear on Linux.

Numpy above 1.22 raises an error regarding slicing and stitching may fail with version 1.23+ until a future update fixes this recent numpy behaviour change.

If installation fails with one specific miniconda installer, a previous version can be used.

Try to avoid using **pip install** along **conda install** package management system.