
easypipe Documentation

Release 1.1.0

Muriel Gelin

July 27, 2021

1	Getting started (Linux)	3
1.1	Check your python version	3
1.2	Check if pip is installed	3
1.3	Software requirements	4
1.3.1	Phenix	4
1.3.2	Open Babel	4
1.4	Download and install easypipe package	4
1.5	Uninstall easypipe package	5
1.6	References	5
2	Getting started (Windows)	7
2.1	Check your python version	7
2.2	Software requirements	7
2.2.1	Phenix	8
2.2.2	Open Babel	8
2.3	Download and install easypipe package	8
2.4	Uninstall easypipe package	9
2.5	References	9
3	easYGet	11
3.1	easYGet usage	11
3.2	What does it do ?	12
4	easYPipe	15
4.1	easYPipe usage	15
4.1.1	References	16
5	easYPipe quickstart guide	17
5.1	1. Retrieve and organize your processed data	17
5.2	2. Prepare the data with ‘prep’	17
5.3	3. Reindex if necessary with ‘reindex’	17
5.4	4. Add ligands with ‘ligands’	18
5.5	5. Process the data with ‘launch’	18
5.5.1	Mode	18
5.5.2	Ligand search	18
5.5.3	Datasets to treat	19
5.6	6. Compile results in a summary file	19
6	easYPipe ‘prep’	21
6.1	Usage	21
6.2	How the data should be organized ?	21

6.3	What does it do ?	22
6.4	References	26
7	easYPipe ‘reindex’	27
7.1	Usage	27
7.2	What does it do ?	27
7.3	References	29
8	easYPipe ‘ligands’	31
8.1	Usage	31
8.2	What does it do ?	31
8.3	References	33
9	easYPipe ‘launch’	35
9.1	Usage	35
9.2	What does it do ?	36
9.2.1	1. Sort mtz files according to space group in reference pdb, and decreasing completeness . .	36
9.2.2	2. List mtz files according to option ‘best’ or ‘autoproc’	36
9.2.3	3. List mtz files with mode and ligand information for running Phenix	37
9.2.4	4. Launch Phenix according to chosen mode and options	41
9.2.5	5. Write results	42
9.3	Phenix options according to modes (only for information)	45
9.4	References	46
10	easYPipe ‘summary’	47
10.1	Usage	47
10.2	What does it do ?	47
11	easYPipe ‘pandda’	49
11.1	Usage	49
11.2	What does it do ?	49
11.3	References	50

You can also grab a hardcopy of the easYPipe documentation in PDF

GETTING STARTED (LINUX)

Check your python version

First, open a terminal.

easYPipe needs at least Python 3.6.

You may already have [Python](#) installed – you can check which version by doing:

```
$ python --version
```

If the version is < 3.0, do:

```
$ python3 --version
```

If that fails or Python 3 version is <3.6, install Python 3.6 using the package manager of your distribution. For example for Ubuntu 16.10 or newer:

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt update
$ sudo apt-get install python3.6
```

If you've updated your Python version from 3.5 to 3.6, you can change default Python 3 to Python 3.6:

```
$ update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.5 1
$ update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.6 2
```

Check that Python 3 has been correctly installed:

```
$ python3 --version
Python 3.6
```

Check if pip is installed

[pip](#) is a package management tool for Python.

Check if pip3 is installed:

```
$ which pip3
```

If that fails, install pip3 using the package manager of your distribution. For example for Ubuntu:

```
$ sudo apt-get install python3-pip
```

Check if pip3 is linked to Python >= 3.6:

```
$ pip3 --version
pip 21.0.1 from /home/username/.local/lib/python3.6/site-packages/pip (python 3.6)
```

If it's not the case, you can try:

```
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python3 get-pip.py
then check:
$ pip3 --version
```

Software requirements

easYPipe is a pipeline that requires other software.

Phenix

You need [Phenix](https://www.phenix-online.org/)¹ installed and to modify PHENIX_PATH value in config.py file accordingly.

Example for PHENIX_PATH in config.py:

```
PHENIX_PATH = "/usr/local/phenix-1.18.2-3874/"
```

Open Babel

[Open Babel](http://openbabel.org/wiki/Main_Page)² is a chemical toolbox needed for ligands.

Example for BABEL_PATH in config.py:

```
BABEL_PATH = "/usr/local/OPENBABEL/openbabel-2-4-1/build/bin/"
```

Important: paths to these software have to be modified in config.py file accordingly to your installation (see next step)

Download and install easypipe package

Download the zip archive that should look like easypipe-1.0.0.tar.gz.

Go where is the archive and unzip:

```
$ tar -xvzf easypipe-1.1.0.tar.gz
```

Go to folder easypipe-1.1.0:

```
$ cd easypipe-1.1.0
```

¹ <https://www.phenix-online.org/>

² http://openbabel.org/wiki/Main_Page

Modify (open in a text editor) **config.py** file (in easypipe-1.1.0 folder), in particular ‘Software used by modules’ part since nothing will work without links to *software*.

Warning: **config.py** file modifications have to be done before installation !! If you want to modify it afterwards, just launch installation again.

Then install:

```
$ sudo python setup.py install # if your default python version is >=3.6
or
$ sudo python3 setup.py install # if your python3 version is >=3.6
or
$ sudo python3.6 setup.py install # if your python3 version is linked to python3.5
```

If you’ve updated you Python version from 3.5 to 3.6, you may encountered some dependencies and conflict problems. Read carefully the error messages. These commands can help you:

```
$ sudo apt-get install libffi-dev

$ pip3 uninstall PyNaCl
$ pip3 install PyNaCl

$ pip3 install setuptools_rust

$ cd /usr/lib/python3/dist-packages
$ sudo ln -s apt_pkg.cpython-{35m,36m}-x86_64-linux-gnu.so
```

Uninstall easypipe package

To uninstall easypipe package properly, do:

```
$ sudo pip3 uninstall easypipe
or
$ sudo python3 -m pip uninstall easypipe
or
$ sudo python3.6 -m pip uninstall easypipe
```

Note: ‘pip3 freeze’ lists the names of all python packages installed.

References

GETTING STARTED (WINDOWS)

Check your python version

First, open a Command Prompt.

Note: To open Command Prompt, simply type 'cmd' in the search box of the Windows taskbar.

easYPipe needs at least Python 3.6.

You may already have [Python](#) installed – you can check which version by doing:

```
C:\> python --version
or
C:\> py -V
```

If Python version is <3.6, install the latest Python 3 version. Download the latest python version [Python downloads site](#). You can follow instructions [here](#). But what you need is simply:

- Run the installer. You can do so by double-clicking python-<version>.exe in your Downloads folder.
- Check the box next to "Add Python <version> to PATH." It's at the bottom of the window.
- Install Now (Customize installation not compulsory).
- Click Disable path length limit. This ensures that Python (and other apps) to use paths more than 2

Check that Python 3 has been correctly installed:

```
C:\> python --version
Python 3.9

or

C:\> py -V
Python 3.9
```

Software requirements

easYPipe is a pipeline that requires other software.

Phenix

You need [Phenix](https://www.phenix-online.org/)¹ installed and to modify PHENIX_PATH value in config.py file accordingly.

Example for PHENIX_PATH in config.py:

```
PHENIX_PATH = r"C:\Users\myname\Phenix\phenix-installer-1.19.1-4122-intel-windows-x86_64"
```

Open Babel

[Open Babel](http://openbabel.org/wiki/Main_Page)² is a chemical toolbox needed for ligands.

Install Open Babel GUI for Windows.

Example for BABEL_PATH in config.py:

```
BABEL_PATH = r"C:\Program Files\OpenBabel-3.1.1"
```

Important: paths to these software have to be modified in config.py file accordingly to your installation (see next step)

Tip: it might be wise to use [LibreOffice](https://www.libreoffice.org/)³ as many csv spreadsheets are generated by easYPipe and Excel does not handle the official csv format (i.e. comma-separated values) well for non-English speaking versions

Download and install easypipe package

Download the zip archive that should look like easypipe-1.1.0.tar.gz.

Go where is the archive and unzip:

```
C:\> tar -xvzf easypipe-1.1.0.tar.gz
```

Go to folder easypipe-1.1.0:

```
C:\> cd easypipe-1.1.0
```

Modify (open with a text editor like Notepad) **config.py** file (in easypipe-1.1.0 folder), in particular ‘Software used by modules’ part since nothing will work without links to *software*.

Warning: **config.py** file modifications have to be done before installation !! If you want to modify it afterwards, just launch installation again.

Then install:

```
C:\> python setup.py install
or
C:\> py setup.py install
```

¹ <https://www.phenix-online.org/>

² http://openbabel.org/wiki/Main_Page

³ <https://www.libreoffice.org/>

Uninstall easypipe package

To uninstall easypipe package properly, do:

```
C:\> pip uninstall easypipe
```

Note: 'pip freeze' lists the names of all python packages installed.

References

EASYGET

easYGet comes with *easYPipe*.

easYGet makes it possible to download all at once the mx 'PROCESSED_DATA' from a synchrotron for a given beamline (several dates possible) and a given acronym.

Downloaded processed datasets are organized by dataset folders, ready to launch easypipe.

For the moment, only works for data from Grenoble ESRF synchrotron.

easYGet usage

`easyget.py [-h] [-s {ESRF}] [-l LOGIN] [-w PASSWORD] [-b BEAMLINE] [-d DATES] [-p PROTEIN]`

optional arguments	description
<code>-h, --help</code>	show this help message and exit
<code>-s {ESRF}, --synchrotron {ESRF}</code>	which synchrotron? (default = ERSF)
<code>-l LOGIN, --login LOGIN</code>	synchrotron login (Example: 'mx—')
<code>-w PASSWORD, --password PASSWORD</code>	password for connexion at synchrotron
<code>-b BEAMLINE, --beamline BEAMLINE</code>	name of the beamline
<code>-d DATES [DATES ...], --dates DATES [DATES ...]</code>	date of the run (format AAAAMMJJ) or list of dates for the same run (format AAAAMMJJ AAAAMMJJ)
<code>-p PROTEIN, --protein PROTEIN</code>	protein acronym or space if no acronym (manual collections)
<code>-g GROUP, --group GROUP</code>	group added via MxCube3 as a supplementary folder level before acronym

Information can be provided either in command line or in interactive mode in a terminal, or mix.

For example, you can simply run:

```
$ easyget.py
```

or:

```
$ easyget.py --login mx1000 --password my_passwd
```

or:

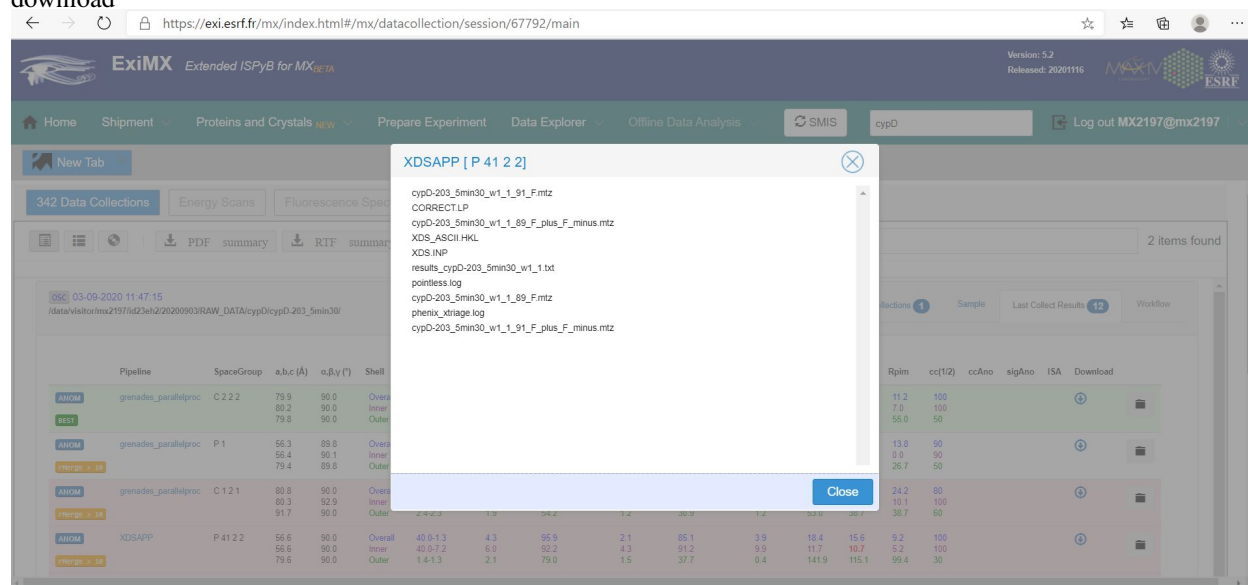
```
$ easyget.py --login mx1000 --password my_passwd --protein myprot --beamline id30a1 --dates 20210131
```

other commands are possible ...

Note: when just one date is given, test also the day after (useful for 24h experiments ...)

What does it do ?

Processed data @ESRF: for each crystal collected, several processes can succeed and for each a zip is available for download



All these processed data are copied this way by easYGet:

- **dataset folder for each dataset**
 - PROC_1, PROC_2, ... if run 1, run 2 ...
 - process sub-folder, with processed data for each process

```

└─ PROCESSED_DATA
  └─ cypD-134_37s
    └─ PROC_1
      └─ autoPROC
        ├── ap_w1_run1_anom_autoPROC.log
        ├── ap_w1_run1_anom_report.pdf
        ├── ap_w1_run1_anom_stارانiso_alldata-unique.mtz
        ├── ap_w1_run1_anom_stارانiso_alldata-unique.stats
        ├── ap_w1_run1_anom_stارانiso_alldata-unique.table1
        ├── ap_w1_run1_anom_summary_inlined.html
        ├── ap_w1_run1_anom_truncate.mtz
        ├── ap_w1_run1_anom_truncate-unique.stats
        ├── ap_w1_run1_anom_truncate-unique.table1
        └── ap_w1_run1_anom_XDS_ASCII.HKL.gz
      └─ XDSAPP
        ├── CORRECT.LP
        ├── cypD-134_37s_w1_1_F.mtz
        ├── cypD-134_37s_w1_1_F_plus_F_minus.mtz
        ├── phenix_xtriage.log
        ├── pointless.log
        ├── results_cypD-134_37s_w1_1.txt
        ├── XDS_ASCII.HKL
        └── XDS.INP
    └─ cypD-172_1min20
      └─ PROC_1
        └─ autoPROC
          ├── ap_w1_run1_anom_autoPROC.log
          ├── ap_w1_run1_anom_report.pdf
          ├── ap_w1_run1_anom_stارانiso_alldata-unique.mtz
          ├── ap_w1_run1_anom_stارانiso_alldata-unique.stats
          ├── ap_w1_run1_anom_stارانiso_alldata-unique.table1
          ├── ap_w1_run1_anom_summary_inlined.html
          ├── ap_w1_run1_anom_truncate.mtz
          ├── ap_w1_run1_anom_truncate-unique.stats
          ├── ap_w1_run1_anom_truncate-unique.table1
          └── ap_w1_run1_anom_XDS_ASCII.HKL.gz
        └─ XDSAPP
          ├── CORRECT.LP
          ├── cypD-172_1min20_w1_1_F.mtz
          └── cypD-172_1min20_w1_1_F_plus_F_minus.mtz

```


EASYPEPIPE

easYPipe is a pipeline for the automatization of ligand and fragment X-Ray crystallographic screening.

easYPipe sequentially runs [phenix.ligand_pipeline](https://www.phenix-online.org/documentation/reference/ligand_pipeline.html)¹ on several datasets of a protein that has been screened with different ligands.

When ligand smiles are given, ligand can be placed automatically in each corresponding structure.

Results are gathered and organized in a 'RESULTS' directory, which facilitates future visualization, refinement and deposition. Corresponding relevant information is summarized in a csv file.

easYPipe is specially adapted for ligand or fragment screening in structure-based drug design projects.

See also:

easYGet that comes with easYPipe, for an automated download of processed data from synchrotron.

easYPipe usage

easypipe.py [-h] data {prep,reindex,ligands,launch,pandda} ...

arguments	description
data	folder with datasets in subfolders (mandatory)
-h, -help	show this help message and exit

Example:

```
$ easypipe.py PROCESSED_DATA -h
```

subcommands	description
<i>prep</i>	prepare data: listing of mtz to treat with information
<i>reindex</i>	try to reindex mtz according to the space group of reference file
<i>ligands</i>	generate ligands files before launching with ligand search
<i>launch</i>	launch all the "phenix.ligand-pipeline" (after the preparation steps)
<i>summary</i>	compile the results of all the 'launch' execution in a single csv file
<i>pandda</i>	copy results from easypipe to a 'PANDDA' folder, as data ready to launch PanDDA

Example:

¹ https://www.phenix-online.org/documentation/reference/ligand_pipeline.html

```
$ easypipe.py PROCESSED_DATA prep
```

You can have a look to the *quickstart guide*.

For a better experience, also read instructions for each subcommands.

References

EASYPEPIPE QUICKSTART GUIDE

1. Retrieve and organize your processed data

Note: You can retrieve your processed data from synchrotron using *easYGet*.

Processed data should be in datasets folders, all grouped in a folder. More information on how to organize your data [here](#).

2. Prepare the data with ‘prep’

The first step is intended to list mtz to be treated:

```
$ easypipe.py PROCESSED_DATA prep
```

where here ‘PROCESSED_DATA’ is the folder with your datasets.

Warning: For Windows users, \$ is the Linux prompt that corresponds to C:> in Windows command prompt, and should not be written

Now, you can have a look at /easypipe/1a_prep/mtz_to_treat_ALL.csv file that lists mtz found in your processed data with information like resolution, completeness or space group.

For more details on this step see [here](#).

3. Reindex if necessary with ‘reindex’

If you see that some mtz should be in higher symmetry space group (in /easypipe/1a_prep/mtz_to_treat_ALL.csv file), then you can try to reindex.

Run:

```
$ easypipe.py PROCESSED_DATA reindex my-ref.mtz
```

where here my-ref.mtz is a reference mtz of your protein.

For more details on this step see [here](#).

4. Add ligands with ‘ligands’

This step is necessary if you want Phenix to try to find and place ligands, or if you want to automatically generate the CIF and PDB of your ligands.

First, you have to fill in the fields <ligand name> and <ligand smiles> of /1c_ligands/ligands_for_datasets.csv file.

Then, run:

```
$ easypipe.py PROCESSED_DATA ligands easYPipe/1c_ligands/ligands_for_datasets_OK.csv
```

where here ligands_for_datasets_OK.csv is the name of your filled ligand csv file.

For more details on this step see [here](#).

5. Process the data with ‘launch’

Now you can run Phenix on your processed mtz.

Mode

Default mode, is ‘fast’ mode. This mode uses rigid body refinement and can be run to get a first result rapidly.

Example:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder
```

where my_ref_folder gather fasta file and pdb files for replacement, and cif file if there is a ligand in the model.

Warning: pdb files should include the row starting with ‘CRYST1’ containing information on space group

Now, have a look at your [results](#) in the corresponding ‘RESULTS’ csv file.

If some processes failed, they probably need longer calculations. You can try ‘full’ mode:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full
```

In case your protein changes its space group, with ligand for example, you can ask not to fix space group. As a result, all mtz could be treated even with ‘bad’ space group. The duration for this will be much longer. But you can only do it for some using simulation mode first (see above):

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode allsg
```

Ligand search

If you want LigandFit to place ligands, you first have to run ‘ligand’ subcommand ([see above](#)).

Then just add ‘-lig’ option:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig
```

The default cutoff for LigandFit to place a ligand is 0.7, but you can change it if you see that it is too high, with ‘-cclig’ option:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --cclig 0.6
```

If several ligands are supposed to fix, you can ask for LigandFit to place more than one ligand, with ‘-nblig’ option:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --cclig 0.6 --nblig 5
```

Datasets to treat

Default behavior is to run phenix.ligand_pipeline on the mtz of best completeness for each dataset, you can start with it.

If there are failures in the treatment of ‘best completeness’ mtz, you can try to treat a higher number of mtz for each dataset.

You can first start by running on mtz from autoPROC process which is generally a good compromise between resolution and completeness:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --autoproc
```

Or you can run on the two first mtz of best completeness for each dataset:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --best 2
```

or more ...:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --best 5
```

If only some datasets are problematic, you can run in simulation mode first, modify the corresponding ‘launch’ csv file in /easYPipe/2_launch/ (replace ‘yes’ by ‘no’ in the ‘to treat’ column, for those not to process), then run again:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --best 5 --simulate
then, after modification of the 'launch' csv file:
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --best 5
```

Then, only selected mtz will be treated, reducing the duration of the treatment.

For more details on this step see [here](#).

6. Compile results in a summary file

If you have run several times ‘launch’ subcommand, you will have several ‘RESULT’ csv files that you probably wish to compile.

Then run:

```
$ easypipe.py PROCESSED_DATA summary
```


EASYPEPIPE ‘PREP’

Important: This step is a first mandatory step for the preparation of the data.

Usage

easypipe.py data prep [-h]

Example:

```
$ easypipe.py PROCESSED_DATA prep
```

How the data should be organized ?

The data folder (whatever it's name) must contain only datasets folders.

Within each dataset folder, the processed data can be organized in several ways:

- a mtz file directly in dataset folder
- a mtz file in a sub-folder, or in a sub-sub-folder ... of dataset folder
- several processes are possible for a dataset, provided that they are in different sub-folders
- if several mtz files are present in the same sub-folder, only one will be treated on the basis of templates (from ESRF EDNA processes)

```
- data folder
  - dataset1
    - file.mtz
    - other-file.log
    :
  - dataset2
    - proc-folder
      - file-direct.mtz
      - other-file.txt
      - other-file.csv
      :
  - dataset3
    - PROC_1
      - autoPROC
        - ap_cypD-122-2min_run1_anom_autoPROC.log
        - ap_cypD-122-2min_run1_anom_report.pdf
        - ap_cypD-122-2min_run1_anom_staraniso_alldata-unique.mtz
        - ap_cypD-122-2min_run1_anom_staraniso_alldata-unique.stats
        - ap_cypD-122-2min_run1_anom_staraniso_alldata-unique.table1
        - ap_cypD-122-2min_run1_anom_summary_inlined.html
        - ap_cypD-122-2min_run1_anom_truncate.mtz
        - ap_cypD-122-2min_run1_anom_truncate-unique.stats
        - ap_cypD-122-2min_run1_anom_truncate-unique.table1
        - ap_cypD-122-2min_run1_anom_XDS_ASCII.HKL.gz
      - another-process
      :
    - PROC_2
      :
```

Note: Data downloaded with *easYGet* are directly in the right tree organization.

What does it do ?

In an ‘easYPipe’ folder created where it is run, ‘prep’ copies each processed data mtz in a sub-folder of the dataset in this way:

- creation of an ‘easYPipe’ treatment directory where it is run
- creation of a subdirectory ‘0_processed_datasets’ where all the datasets folder are created
- creation of a ‘data’ folder in each dataset folder and copy in this folder of processed mtz and log files
- if there are several mtz in a folder, search for ‘EDNA’ treatment template and selects the right mtz file

Then:

- launch of *xtriage* ¹ for each mtz to get resolution, completeness, space group and cell parameters

¹ <https://www.phenix-online.org/documentation/reference/xtriage.html>

```

easYPipe/
├── 0_processed_datasets
│   ├── cypD-134_37s
│   │   ├── data
│   │   │   ├── mtz001_PROC_1_autoPROC
│   │   │   │   ├── ap_w1_run1_anom_autoPROC.log
│   │   │   │   ├── ap_w1_run1_anom_truncate.mtz
│   │   │   │   ├── ap_w1_run1_anom_XDS_ASCII.HKL.gz
│   │   │   │   ├── xtriage_ap_w1_run1_anom_truncate.log
│   │   │   │   ├── xtriage-verbose_labels.log
│   │   │   │   └── xtriage-verbose.log
│   │   │   └── mtz002_PROC_1_XDSAPP
│   │   │       ├── CORRECT.LP
│   │   │       ├── cypD-134_37s_w1_1_F_plus_F_minus.mtz
│   │   │       ├── phenix_xtriage.log
│   │   │       ├── pointless.log
│   │   │       ├── XDS_ASCII.HKL
│   │   │       ├── xtriage_cypD-134_37s_w1_1_F_plus_F_minus.log
│   │   │       ├── xtriage-verbose_labels.log
│   │   │       └── xtriage-verbose.log
│   │   ├── ligand
│   │   └── cypD-172_1min20
│   │       ├── data
│   │       │   ├── mtz001_PROC_1_autoPROC
│   │       │   │   ├── ap_w1_run1_anom_autoPROC.log
│   │       │   │   ├── ap_w1_run1_anom_truncate.mtz
│   │       │   │   ├── ap_w1_run1_anom_XDS_ASCII.HKL.gz
│   │       │   │   ├── xtriage_ap_w1_run1_anom_truncate.log
│   │       │   │   ├── xtriage-verbose_labels.log
│   │       │   │   └── xtriage-verbose.log
│   │       │   └── mtz002_PROC_1_XDSAPP
│   │       │       ├── CORRECT.LP
│   │       │       ├── cypD-172_1min20_w1_1_F_plus_F_minus.mtz
│   │       │       ├── phenix_xtriage.log
│   │       │       ├── pointless.log
│   │       │       ├── XDS_ASCII.HKL
│   │       │       ├── xtriage_cypD-172_1min20_w1_1_F_plus_F_minus.log
│   │       │       ├── xtriage-verbose_labels.log
│   │       │       └── xtriage-verbose.log
│   │       └── mtz003_PROC_1_XIA2_DIALS
│   │           └── di_w1_run1_anom_AUTOMATIC_DEFAULT_aimless.log

```

- information on mtz files to be treated written in '/easypipe/1a_prep/mtz_to_treat_ALL.csv' file

mtz_to_treat_ALL.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Libération Sans 10 G I S A

	A	B	C	D	E	F	G	H
1	dataset	mtz nb	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
2	cypD-134_37s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.31	97.98	P422	56.858, 56.858, 87.565, 90, 90, 90
3	cypD-134_37s	mtz002	PROC_1_XDSAPP	cypD-134_37s_w1_1_F.mtz	1.64	42.53	P2	57.0855, 57.0855, 87.787, 90, 90, 90
4	cypD-172_1min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	4.05	52.83	P1	55.243, 55.313, 79.525, 90, 90, 90
5	cypD-172_1min20	mtz002	PROC_1_XDSAPP	cypD-172_1min20_w1_1_F.mtz	3.88	43.0	P1	55.816, 55.848, 80.31, 89.882, 89.99, 89.867
6	cypD-172_1min20	mtz003	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	3.08	71.63	P21	55.3039, 55.3039, 79.668, 90, 90, 90
7	cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.51	97.88	P422	56.74, 56.74, 80.034, 90, 90, 90
8	cypD-203_5min30	mtz002	PROC_1_XDSAPP	cypD-203_5min30_w1_1_89_F.mtz	1.32	90.39	P422	56.593, 56.593, 79.576, 90, 90, 90
9	cypD-203_5min30	mtz003	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.67	83.63	C2	56.701, 56.701, 79.7425, 90, 90, 90
10	cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 90, 90
11	cypD-248_5min	mtz002	PROC_1_EDNA proc	ep_cypD-248_5min_w1_run1_anom_truncate.mtz	1.0	85.05	P41212	57.264, 57.264, 87.827, 90, 90, 90
12	cypD-248_5min	mtz003	PROC_1_XDSAPP	cypD-248_5min_w1_1_96_F.mtz	1.03	85.68	P43212	57.274, 57.274, 87.837, 90, 90, 90
13	cypD-248_5min	mtz004	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.38	80.19	P2	57.2164, 57.2164, 87.6953, 90, 90, 90
14	cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.9	P422	57.024, 57.024, 87.466, 90, 90, 90
15	cypD-317_2min	mtz002	PROC_1_EDNA proc	ep_cypD-317_2min_w1_run1_anom_truncate.mtz	1.07	95.13	P4212	56.9836, 56.9836, 87.471, 90, 90, 90
16	cypD-317_2min	mtz003	PROC_1_XDSAPP	cypD-317_2min_w1_1_92_F.mtz	1.06	90.43	P41212	57.017, 57.017, 87.46, 90, 90, 90
17	cypD-343_5min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.11	83.81	P41212	57.256, 57.256, 87.603, 90, 90, 90
18	cypD-343_5min20	mtz002	PROC_1_EDNA proc	ep_cypD-343_5min20_w1_run1_anom_truncate.mtz	1.11	83.97	P41212	57.235, 57.235, 87.572, 90, 90, 90
19	cypD-343_5min20	mtz003	PROC_1_XDSAPP	cypD-343_5min20_w1_1_96_F.mtz	1.11	77.88	P43212	57.269, 57.269, 87.62, 90, 90, 90
20	cypD-343_5min20	mtz004	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.44	83.17	C2	80.585, 80.1558, 87.5141, 90, 88.1608, 90
21	cypD-438_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.07	93.24	P422	57.294, 57.294, 87.555, 90, 90, 90
22	cypD-438_1min	mtz002	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.05	82.57	C2221	80.7025, 80.8632, 87.4104, 90, 90, 90
23	cypD-440_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	2.7	57.25	P1	55.3529, 55.3529, 78.971, 90, 90, 90
24	cypD-440_5min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F.mtz	1.62	96.57	P41212	56.512, 56.512, 81.831, 90, 90, 90
25	cypD-619_4min45	mtz001	PROC_1_EDNA proc	ep_cypD-619_4min45_w1_run1_anom_truncate.mtz	1.96	90.86	P422	55.887, 55.887, 79.878, 90, 90, 90
26	cypD-619_4min45	mtz002	PROC_1_XDSAPP	cypD-619_4min45_w1_1_F.mtz	1.96	90.86	P422	55.887, 55.887, 79.878, 90, 90, 90
27	cypD-619_4min45	mtz003	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.69	97.05	P4212	56.4117, 56.4117, 79.9383, 90, 90, 90
28	cypD-860_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	2.86	83.07	P422	57.185, 57.185, 87.586, 90, 90, 90
29	cypD-860_57s	mtz002	PROC_1_EDNA proc	ep_cypD-860_57s_w1_run1_anom_truncate.mtz	1.14	84.49	P41212	57.328, 57.328, 87.346, 90, 90, 90
30	cypD-860_57s	mtz003	PROC_1_XDSAPP	cypD-860_57s_w1_1_20_F.mtz	1.42	68.54	C2221	80.909, 80.921, 87.646, 90, 90, 90
31	cypD-860_57s	mtz004	PROC_1_XIA2 DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.27	91.66	P2221	57.2601, 57.2601, 87.2433, 90, 90, 90

-creation of a csv file ‘/easypipe/1c_ligands/ligands_for_datasets.csv’ for future ligand generation with **eLBOW**²

² <https://www.phenix-online.org/documentation/reference/elbow.html>

ligands_for_datasets.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Libération Sans 10 G I S A

J35

	A	B	C
1	dataset	ligand name	ligand smiles
2	cypD-134_37s		
3	cypD-172_1min20		
4	cypD-203_5min30		
5	cypD-248_5min		
6	cypD-317_2min		
7	cypD-343_5min20		
8	cypD-438_1min		
9	cypD-440_5min		
10	cypD-619_4min45		
11	cypD-860_57s		
12	cypD-861_5min30		
13	cypD-863_2min30		
14	cypD-864_1min		
15	cypD-865_5min		
16	cypD-866_6min		
17	cypD-867_5min30		
18	cypD-869_5min		
19	cypD-872_2min30		
20	cypD-874_5min30		
21	cypD-877_5min		
22	cypD-878_5min		
23	cypD-879_3min20		
24	cypD-880_1min		
25	cypD-881_2min20		
26	cypD-882_5min		
27	cypD-884_5min		
28	cypD-885_55s		
29	cypD-886_5min		
30	cypD-887_4min		
31	cypD-888_1min55		
32	cypD-889_2min		
33	cypD-890_1min30		
34	cypD-891_2min14		

You have to fill 'ligand name' and 'ligand smiles' fields before running '*easYPipe ligands subcommand*'.

Caution: Save the modified csv file somewhere else or with another name if you don't want to overwrite it in case you launch 'prep' sub-command again ...

You can also run '*easYPipe reindex subcommand*' if some mtz should be in higher symmetry space group.

If you are not interested in ligand placement or reindexation, you can directly run '*easYPipe launch subcommand*'.

References

EASYPEPIPE ‘REINDEX’

This optional step is useful when several mtz should be in higher symmetry space group.

The program try to reindex according to the space group of the reference mtz.

Example: P422 can be re-indexed to P41212.

Usage

easypipe.py data reindex [-h] ref_mtz

arguments	description
-h, --help	show this help message and exit
sg_ref	space group of reference for reindexing (name or number)

Example:

```
$ easypipe.py PROCESSED_DATA reindex P41212
$ easypipe.py PROCESSED_DATA reindex 92
```

What does it do ?

- try to reindex mtz file with [reflection_file_converter](https://phenix-online.org/documentation/reference/reflection_file_tools.html)¹ if space group is different from reference space group

¹ https://phenix-online.org/documentation/reference/reflection_file_tools.html

```

~~~~~
~
SUMMARY
~
~~~~~
mtz candidates for reindexing (space group different from reference file): 93

cypD-134_37s/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate.mtz / P422
cypD-134_37s/mtz002_PROC_1_XDSAPP/cypD-134_37s_wl_1_F.mtz / P2
cypD-172_1min20/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate.mtz / P1
cypD-172_1min20/mtz002_PROC_1_XDSAPP/cypD-172_1min20_wl_1_F.mtz / P1
cypD-172_1min20/mtz003_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free.mtz / P21
cypD-203_5min30/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate.mtz / P422
cypD-203_5min30/mtz002_PROC_1_XDSAPP/cypD-203_5min30_wl_1_89_F.mtz / P422
cypD-203_5min30/mtz003_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free.mtz / C2
cypD-248_5min/mtz003_PROC_1_XDSAPP/cypD-248_5min_wl_1_96_F.mtz / P43212
cypD-248_5min/mtz004_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free.mtz / P2
cypD-317_2min/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate.mtz / P422
cypD-317_2min/mtz002_PROC_1_EDNA_proc/ep_cypD-317_2min_wl_run1_anom_truncate.mtz / P4212
cypD-343_5min20/mtz003_PROC_1_XDSAPP/cypD-343_5min20_wl_1_96_F.mtz / P43212
cypD-343_5min20/mtz004_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free.mtz / C2
cypD-438_1min/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate.mtz / P422
cypD-438_1min/mtz002_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free.mtz / C2221
cypD-440_5min/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate.mtz / P1
cypD-619_4min45/mtz001_PROC_1_EDNA_proc/ep_cypD-619_4min45_wl_run1_anom_truncate.mtz / P422
cypD-619_4min45/mtz002_PROC_1_XDSAPP/cypD-619_4min45_wl_1_F.mtz / P422
cypD-619_4min45/mtz003_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free.mtz / P4212

Number of reindexed mtz: 51
cypD-134_37s/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-203_5min30/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-203_5min30/mtz002_PROC_1_XDSAPP/cypD-203_5min30_wl_1_89_F_reindexed_P41212.mtz
cypD-248_5min/mtz003_PROC_1_XDSAPP/cypD-248_5min_wl_1_96_F_reindexed_P41212.mtz
cypD-317_2min/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-317_2min/mtz002_PROC_1_EDNA_proc/ep_cypD-317_2min_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-343_5min20/mtz003_PROC_1_XDSAPP/cypD-343_5min20_wl_1_96_F_reindexed_P41212.mtz
cypD-438_1min/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-619_4min45/mtz001_PROC_1_EDNA_proc/ep_cypD-619_4min45_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-619_4min45/mtz003_PROC_1_XIA2_DIALS/di_wl_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz
cypD-860_57s/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-863_2min30/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-864_1min/mtz001_PROC_1_autoPROC/ap_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-864_1min/mtz002_PROC_1_EDNA_proc/ep_cypD-864_1min_wl_run1_anom_truncate_reindexed_P41212.mtz
cypD-864_1min/mtz003_PROC_1_XDSAPP/cypD-864_1min_wl_1_95_F_reindexed_P41212.mtz
cypD-865_5min/mtz003_PROC_1_XDSAPP/cypD-865_5min_wl_1_96_F_reindexed_P41212.mtz
cypD-866_6min/mtz002_PROC_1_EDNA_proc/ep_cypD-866_6min_wl_run1_anom_truncate_reindexed_P41212.mtz

```

- launch `xtrriage`² for each successfully reindexed mtz to get resolution, completeness, space group and cell parameters
- write a new ‘mtz_to_treat_ALL_reindexed.csv’ in in ‘easypipe/1b_reindex...’ folder, with reindexed mtz files information

² <https://www.phenix-online.org/documentation/reference/xtrriage.html>

mtz_to_treat_ALL.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Liberation Sans 10 G I S A

	A	B	C	D	E	F	G	H
1	dataset	mtz nb	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
2	cypD-134_37s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.31	97.98	P422	56.858, 56.858, 87.565, 90, 90, 90
3	cypD-134_37s	mtz002	PROC_1_XDSAPP	cypD-134_37s_w1_1_F.mtz	1.64	42.53	P2	57.0855, 57.0855, 87.787, 90, 90, 90
4	cypD-172_1min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	4.05	52.83	P1	55.243, 55.313, 79.525, 90, 90, 90
5	cypD-172_1min20	mtz002	PROC_1_XDSAPP	cypD-172_1min20_w1_1_F.mtz	3.88	43.0	P1	55.816, 55.848, 80.31, 89.882, 89.99, 89.867
6	cypD-172_1min20	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	3.08	71.63	P21	55.3039, 55.3039, 79.668, 90, 90, 90
7	cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.51	97.88	P422	56.74, 56.74, 80.034, 90, 90, 90
8	cypD-203_5min30	mtz002	PROC_1_XDSAPP	cypD-203_5min30_w1_1_89_F.mtz	1.32	90.39	P422	56.593, 56.593, 79.576, 90, 90, 90
9	cypD-203_5min30	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.67	83.63	C2	56.701, 56.701, 79.7425, 90, 90, 90
10	cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 90, 90
11	cypD-248_5min	mtz002	PROC_1_EDNA_proc	ep_cypD-248_5min_w1_run1_anom_truncate.mtz	1.0	85.05	P41212	57.264, 57.264, 87.827, 90, 90, 90
12	cypD-248_5min	mtz003	PROC_1_XDSAPP	cypD-248_5min_w1_1_96_F.mtz	1.03	85.68	P43212	57.274, 57.274, 87.872, 90, 90, 90
13	cypD-248_5min	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.38	80.19	P2	57.2164, 57.2164, 87.6953, 90, 90, 90
14	cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.9	P422	57.024, 57.024, 87.466, 90, 90, 90
15	cypD-317_2min	mtz002	PROC_1_EDNA_proc	ep_cypD-317_2min_w1_run1_anom_truncate.mtz	1.07	95.13	P4212	56.9836, 56.9836, 87.471, 90, 90, 90
16	cypD-317_2min	mtz003	PROC_1_XDSAPP	cypD-317_2min_w1_1_92_F.mtz	1.06	90.43	P41212	57.017, 57.017, 87.46, 90, 90, 90
17	cypD-343_5min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.11	83.81	P41212	57.256, 57.256, 87.603, 90, 90, 90
18	cypD-343_5min20	mtz002	PROC_1_EDNA_proc	ep_cypD-343_5min20_w1_run1_anom_truncate.mtz	1.11	83.97	P41212	57.235, 57.235, 87.572, 90, 90, 90
19	cypD-343_5min20	mtz003	PROC_1_XDSAPP	cypD-343_5min20_w1_1_96_F.mtz	1.11	77.88	P43212	57.269, 57.269, 87.62, 90, 90, 90
20	cypD-343_5min20	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.44	83.17	C2	80.585, 80.1558, 87.5141, 90, 88.1608, 90
21	cypD-438_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.07	93.24	P422	57.294, 57.294, 87.555, 90, 90, 90
22	cypD-438_1min	mtz002	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.05	82.57	C2221	80.7025, 80.8632, 87.4104, 90, 90, 90
23	cypD-440_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	2.7	57.25	P1	55.3529, 55.3529, 78.971, 90, 90, 90
24	cypD-440_5min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F.mtz	1.62	96.57	P41212	56.512, 56.512, 81.831, 90, 90, 90
25	cypD-619_4min45	mtz001	PROC_1_EDNA_proc	ep_cypD-619_4min45_w1_run1_anom_truncate.mtz	1.96	90.86	P422	55.887, 55.887, 79.878, 90, 90, 90
26	cypD-619_4min45	mtz002	PROC_1_XDSAPP	cypD-619_4min45_w1_1_F.mtz	1.96	90.86	P422	55.887, 55.887, 79.878, 90, 90, 90
27	cypD-619_4min45	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.69	97.05	P4212	56.4117, 56.4117, 79.9383, 90, 90, 90
28	cypD-860_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	2.86	83.07	P422	57.185, 57.185, 87.586, 90, 90, 90
29	cypD-860_57s	mtz002	PROC_1_EDNA_proc	ep_cypD-860_57s_w1_run1_anom_truncate.mtz	1.14	84.49	P41212	57.328, 57.328, 87.346, 90, 90, 90
30	cypD-860_57s	mtz003	PROC_1_XDSAPP	cypD-860_57s_w1_1_20_F.mtz	1.42	68.54	C2221	80.909, 80.921, 87.646, 90, 90, 90
31	cypD-860_57s	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.27	91.66	P2221	57.2601, 57.2601, 87.2433, 90, 90, 90

becomes:

mtz_to_treat_ALL_reindexed.csv - LibreOffice Calc <2>

Fichier Édition Affichage Insertion Format Feuille Données Outils Fenêtre Aide

Liberation Sans 10 B I U

	A	B	C	D	E	F	G	H
1	dataset	mtz nb	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
2	cypD-134_37s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.31	98.12	P41212	56.858, 56.858, 87.565, 90, 90, 90
3	cypD-134_37s	mtz002	PROC_1_XDSAPP	cypD-134_37s_w1_1_F_plus_F_minus_reindexed_P41212.mtz	1.64	97.33	P41212	57.0855, 57.0855, 87.787, 90, 90, 90
4	cypD-172_1min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	4.05	96.94	P41212	55.243, 55.313, 79.525, 90, 90, 90
5	cypD-172_1min20	mtz002	PROC_1_XDSAPP	cypD-172_1min20_w1_1_F_plus_F_minus_reindexed_P41212.mtz	3.88	98.4	P41212	55.816, 55.848, 80.31, 89.882, 89.99, 89.867
6	cypD-172_1min20	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	3.08	76.05	P41212	55.3039, 55.3039, 79.668, 90, 90, 90
7	cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.51	98.1	P41212	56.74, 56.74, 80.034, 90, 90, 90
8	cypD-203_5min30	mtz002	PROC_1_XDSAPP	cypD-203_5min30_w1_1_89_F_plus_F_minus_reindexed_P41212.mtz	1.32	90.47	P41212	56.593, 56.593, 79.576, 90, 90, 90
9	cypD-203_5min30	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.67	48.67	P41212	56.701, 56.701, 79.7425, 90, 90, 90
10	cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 90, 90
11	cypD-248_5min	mtz002	PROC_1_EDNA_proc	ep_cypD-248_5min_w1_run1_anom_truncate.mtz	1.0	85.05	P41212	57.264, 57.264, 87.827, 90, 90, 90
12	cypD-248_5min	mtz003	PROC_1_XDSAPP	cypD-248_5min_w1_1_92_F_plus_F_minus.mtz	1.03	91.47	P41212	57.274, 57.274, 87.837, 90, 90, 90
13	cypD-248_5min	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.38	97.92	P41212	57.2164, 57.2164, 87.6953, 90, 90, 90
14	cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.14	98.99	P41212	57.024, 57.024, 87.466, 90, 90, 90
15	cypD-317_2min	mtz002	PROC_1_EDNA_proc	ep_cypD-317_2min_w1_run1_anom_truncate_reindexed_P41212.mtz	1.07	90.05	P41212	56.9836, 56.9836, 87.471, 90, 90, 90
16	cypD-317_2min	mtz003	PROC_1_XDSAPP	cypD-317_2min_w1_1_96_F_plus_F_minus_reindexed_P41212.mtz	1.06	90.42	P41212	57.017, 57.017, 87.46, 90, 90, 90
17	cypD-343_5min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.11	83.81	P41212	57.256, 57.256, 87.603, 90, 90, 90
18	cypD-343_5min20	mtz002	PROC_1_EDNA_proc	ep_cypD-343_5min20_w1_run1_anom_truncate.mtz	1.11	83.97	P41212	57.235, 57.235, 87.572, 90, 90, 90
19	cypD-343_5min20	mtz003	PROC_1_XDSAPP	cypD-343_5min20_w1_1_96_F_plus_F_minus_reindexed_P41212.mtz	1.11	77.88	P41212	57.269, 57.269, 87.62, 90, 90, 90
20	cypD-343_5min20	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.44	48.32	P41212	80.585, 80.1558, 87.5141, 90, 88.1608, 90
21	cypD-438_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.07	93.27	P41212	57.294, 57.294, 87.555, 90, 90, 90
22	cypD-438_1min	mtz002	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.05	44.52	P41212	80.7025, 80.8632, 87.4104, 90, 90, 90
23	cypD-440_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	2.7	94.39	P41212	55.3529, 55.3529, 78.971, 90, 90, 90
24	cypD-440_5min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F_plus_F_minus.mtz	1.62	97.1	P41212	56.512, 56.512, 81.831, 90, 90, 90
25	cypD-619_4min45	mtz001	PROC_1_EDNA_proc	ep_cypD-619_4min45_w1_run1_anom_truncate_reindexed_P41212.mtz	1.96	89.87	P41212	55.887, 55.887, 79.878, 90, 90, 90
26	cypD-619_4min45	mtz002	PROC_1_XDSAPP	cypD-619_4min45_w1_1_F_plus_F_minus.mtz	1.96	90.86	P422	55.887, 55.887, 79.878, 90, 90, 90
27	cypD-619_4min45	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.69	96.62	P41212	56.4117, 56.4117, 79.9383, 90, 90, 90
28	cypD-860_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	2.86	83.76	P41212	57.185, 57.185, 87.586, 90, 90, 90
29	cypD-860_57s	mtz002	PROC_1_EDNA_proc	ep_cypD-860_57s_w1_run1_anom_truncate.mtz	1.14	84.49	P41212	57.328, 57.328, 87.346, 90, 90, 90

where P422, P41212 processed data have been successfully reindexed to P41212 space group.

Warning: If you run again ‘prep’ step for any reason like adding new datasets, you will have to run again this ‘reindex’ step. Even if they don’t need to be reindexed, you have to run ‘reindex’ step to have the right reindexed csv file including these new datasets.

References

EASYPEPIPE 'LIGANDS'

This step is mandatory if you want Phenix to search ligand, else it is optional.

'ligands' subcommand generates pdb and cif from smiles with [eLBOW](#)².

Important: First, template csv file generated with 'prep' subcommand have to be completed with ligands names and smiles.

Usage

easypipe.py data ligands [-h] ligands_csv

arguments	description
-h, --help	show this help message and exit
ligands_csv	ligands_for_datasets.csv file from 'prep' with ligands names and smiles completed (mandatory)

Example:

```
$ easypipe.py PROCESSED_DATA ligands easYPipe/1c_ligands/ligands_for_datasets_OK.csv
```

What does it do ?

- First, you have to fill in the fields 'ligand name' and 'ligand smiles' of /1c_ligands/ligands_for_datasets.csv csv file, then save the csv file somewhere else or with another name if you don't want to overwrite it in case you run 'prep' subcommand again ...

² <https://www.phenix-online.org/documentation/reference/elbow.html>

ligands_for_datasets_OK.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

L34

	A	B	C
1	dataset	ligand name	ligand smiles
2	cypD-134_37s		
3	cypD-172_1min20	172	<chem>Nc1cccc(c1)C(O)=O</chem>
4	cypD-203_5min30		
5	cypD-317_2min	317	<chem>c1cc2cccnc2[nH]1</chem>
6	cypD-343_5min20	343	<chem>NS(=O)(=O)c1ccccc1</chem>
7	cypD-438_1min	438	<chem>c1n[nH]c2ccccc12</chem>
8	cypD-440_5min	440	<chem>C1Cc2ccccc2N1</chem>
9	cypD-610_4min45	610	<chem>OB(O)c1ccccc1</chem>

- ‘ligands’ subcommand generates pdb and cif of ligands and copies them in corresponding processed dataset folders, in a ‘ligand’ folder. It first creates a smiles file accordingly to the ligands_csv input, canonizes it thanks to [Open Babel](http://openbabel.org/wiki/Main_Page)¹ and converts it with [eLBOW](#)² to pdb and cif.

¹ http://openbabel.org/wiki/Main_Page

```

0_processed_datasets/
├── cypD-134_37s
│   ├── data
│   │   ├── mtz001_PROC_1_autoPROC
│   │   │   ├── ap_w1_run1_anom_truncate.mtz
│   │   │   ├── ap_w1_run1_anom_truncate.mtz.old
│   │   │   ├── ap_w1_run1_anom_truncate_reindexed_P41212.mtz
│   │   │   ├── xtriage_ap_w1_run1_anom_truncate.log
│   │   │   ├── xtriage-verbose_labels.log
│   │   │   └── xtriage-verbose.log
│   │   └── mtz002_PROC_1_XDSAPP
│   │       ├── cypD-134_37s_w1_1_F_plus_F_minus.mtz
│   │       ├── xtriage_cypD-134_37s_w1_1_F_plus_F_minus.log
│   │       ├── xtriage-verbose_labels.log
│   │       └── xtriage-verbose.log
│   └── ligand
├── cypD-172_1min20
│   ├── data
│   │   ├── mtz001_PROC_1_autoPROC
│   │   │   ├── ap_w1_run1_anom_truncate.mtz
│   │   │   ├── xtriage_ap_w1_run1_anom_truncate.log
│   │   │   ├── xtriage-verbose_labels.log
│   │   │   └── xtriage-verbose.log
│   │   ├── mtz002_PROC_1_XDSAPP
│   │   │   ├── cypD-172_1min20_w1_1_F_plus_F_minus.mtz
│   │   │   ├── xtriage_cypD-172_1min20_w1_1_F_plus_F_minus.log
│   │   │   ├── xtriage-verbose_labels.log
│   │   │   └── xtriage-verbose.log
│   │   └── mtz003_PROC_1_XIA2_DIALS
│   │       ├── di_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz
│   │       ├── xtriage_di_w1_run1_anom_AUTOMATIC_DEFAULT_free.log
│   │       ├── xtriage-verbose_labels.log
│   │       └── xtriage-verbose.log
│   └── ligand
│       ├── 172.cif
│       ├── 172.elbow_opt.xyz
│       ├── 172.options.pickle
│       ├── 172.pdb
│       ├── 172.pickle
│       └── 172.smi
└── cypD-203_5min30

```

Important: If `eLBOW`² fails to generate pdb and cif from smiles, you can copy your own cif in ligand sub-folder of the corresponding dataset, `eLBOW`² will generate pdb from this cif.

References

EASYPEPIPE ‘LAUNCH’

‘launch’ subcommand runs [phenix.ligand_pipeline](https://www.phenix-online.org/documentation/reference/ligand_pipeline.html)¹ on all the mtz (several processed data, several datasets) according to options and information in ‘mtz_to_treat_ALL.csv’ file.

Usage

easypipe.py data launch [-h] [-m {fast,full,allsg}] [-l] [-n NUMBER] [-c NUMBER] [-b NUMBER | -a] [-s] [-t TEMPLATE] ref

arguments	description
ref	folder with fasta file and pdb file for replacement, and cif(s) if ligand(s) in the model

Warning: reference pdb files should include the row starting with ‘CRYST1’ containing information on space group

optional arguments	description
-h, --help	show this help message and exit
-m {fast,full,allsg}, --mode {fast,full,allsg}	running mode: fast, full, or allsg (default = fast)
-l, --lig	for ligand search and placement
-n NUMBER, --nblig NUMBER	number of ligand copies to be searched (default = 1, max 9 for the moment).
-c NUMBER, --cclig NUMBER	minimum CC to consider a ligand placement correct (default = 0.7). Ligands with at least this CC will be incorporated into the current model for refinement.
-b NUMBER, --best NUMBER	launch only for mtz with best completeness, NUMBER indicates how many mtz to treat (default 1), ex: --best 2
-a, --autoproc	launch only for mtz from autoPROC, or if none launch for mtz with best completeness
-s, --simulate	only simulate, generate a csv file according to the future launch options. Give the possibility to modify the csv file to choose not to launch certain treatments, before restarting without simulation mode.
-t TEMPLATE, --template TEMPLATE	optional template name for log files and result folders, in case re-launching with different reference pdb of the same space group (else will overwrite).

Example:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --best 2 --cclig 0.6
equivalent to:
$ easypipe.py PROCESSED_DATA launch my_ref_folder -m full -l -b 2 -c 0.6
```

¹ https://www.phenix-online.org/documentation/reference/ligand_pipeline.html

What does it do ?

1. Sort mtz files according to space group in reference pdb, and decreasing completeness

dataset	mtz nb	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
cypD-134_37s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.31	98.12	P41212	56.858, 56.858, 87.565, 90, 90, 90
cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.51	98.1	P41212	56.74, 56.74, 80.034, 90, 90, 90
cypD-203_5min30	mtz002	PROC_1_XDSAPP	cypD-203_5min30_w1_1_89_F_reindexed_P41212.mtz	1.32	90.47	P41212	56.593, 56.593, 79.576, 90, 90, 90
cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 90, 90
cypD-248_5min	mtz003	PROC_1_XDSAPP	cypD-248_5min_w1_1_96_F_reindexed_P41212.mtz	1.03	85.68	P41212	57.274, 57.274, 87.837, 90, 90, 90
cypD-248_5min	mtz002	PROC_1_EDNA_proc	ep_cypD-248_5min_w1_run1_anom_truncate.mtz	1.0	85.05	P41212	57.264, 57.264, 87.827, 90, 90, 90
cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.14	98.99	P41212	57.024, 57.024, 87.466, 90, 90, 90
cypD-317_2min	mtz002	PROC_1_EDNA_proc	ep_cypD-317_2min_w1_run1_anom_truncate_reindexed_P41212.mtz	1.07	95.21	P41212	56.9836, 56.9836, 87.471, 90, 90, 90
cypD-317_2min	mtz003	PROC_1_XDSAPP	cypD-317_2min_w1_1_92_F.mtz	1.06	90.43	P41212	57.017, 57.017, 87.46, 90, 90, 90
cypD-343_5min20	mtz002	PROC_1_EDNA_proc	ep_cypD-343_5min20_w1_run1_anom_truncate.mtz	1.11	83.97	P41212	57.235, 57.235, 87.572, 90, 90, 90
cypD-343_5min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.11	83.81	P41212	57.256, 57.256, 87.603, 90, 90, 90
cypD-343_5min20	mtz003	PROC_1_XDSAPP	cypD-343_5min20_w1_1_96_F_reindexed_P41212.mtz	1.11	77.88	P41212	57.269, 57.269, 87.62, 90, 90, 90
cypD-438_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.07	93.27	P41212	57.294, 57.294, 87.555, 90, 90, 90
cypD-440_5min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F.mtz	1.62	96.57	P41212	56.512, 56.512, 81.831, 90, 90, 90
cypD-619_4min45	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.69	97.29	P41212	56.4117, 56.4117, 79.9383, 90, 90, 90
cypD-619_4min45	mtz001	PROC_1_EDNA_proc	ep_cypD-619_4min45_w1_run1_anom_truncate_reindexed_P41212.mtz	1.96	91.14	P41212	55.887, 55.887, 79.878, 90, 90, 90
cypD-860_57s	mtz002	PROC_1_EDNA_proc	ep_cypD-860_57s_w1_run1_anom_truncate.mtz	1.14	84.49	P41212	57.328, 57.328, 87.346, 90, 90, 90
cypD-860_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.89	99.95	P41212	57.402, 57.402, 88.423, 90, 90, 90
cypD-863_2min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.24	94.96	P41212	56.917, 56.917, 86.828, 90, 90, 90
cypD-864_1min	mtz003	PROC_1_XDSAPP	cypD-864_1min_w1_1_95_F_plus_F_minus_reindexed_P41212.mtz	1.36	97.99	P41212	55.835, 55.835, 80.146, 90, 90, 90
cypD-865_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.1	97.94	P41212	57.319, 57.319, 87.742, 90, 90, 90
cypD-866_6min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.05	81.21	P41212	57.231, 57.231, 87.54, 90, 90, 90
cypD-867_5min30	mtz003	PROC_1_XDSAPP	cypD-867_5min30_w1_1_92_F_plus_F_minus.mtz	1.06	95.51	P41212	57.312, 57.312, 87.716, 90, 90, 90
cypD-869_5min	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.42	97.96	P41212	56.909, 56.909, 87.2498, 90, 90, 90
cypD-872_2min30	mtz002	PROC_1_EDNA_proc	ep_cypD-872_2min30_w1_run1_anom_truncate_reindexed_P41212.mtz	2.32	95.81	P41212	55.673, 55.673, 80.398, 90, 90, 90
cypD-875_4min30	mtz002	PROC_1_XDSAPP	cypD-875_4min30_w1_1_92_F_plus_F_minus.mtz	1.42	96.72	P41212	55.502, 55.502, 85.63, 90, 90, 90

If there are datasets without any mtz to treat according to space group, these datasets are listed in another csv file (“datasets_without_mtz_<sg_ref>.csv”).

2. List mtz files according to option ‘best’ or ‘autoproc’

- Option example: –best 1 (default)

List only mtz with best completeness for each dataset.

dataset	mtz nb	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
cypD-134_37s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.31	98.12	P41212	56.858, 56.858, 87.565, 90, 90, 90
cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.51	98.1	P41212	56.74, 56.74, 80.034, 90, 90, 90
cypD-203_5min30	mtz002	PROC_1_XDSAPP	cypD-203_5min30_w1_1_89_F_reindexed_P41212.mtz	1.32	90.47	P41212	56.593, 56.593, 79.576, 90, 90, 90
cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 90, 90
cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.14	98.99	P41212	57.024, 57.024, 87.466, 90, 90, 90
cypD-343_5min20	mtz002	PROC_1_EDNA_proc	ep_cypD-343_5min20_w1_run1_anom_truncate.mtz	1.11	83.97	P41212	57.235, 57.235, 87.572, 90, 90, 90
cypD-440_5min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F_plus_F_minus.mtz	1.62	97.1	P41212	56.512, 56.512, 81.831, 90, 90, 90
cypD-619_4min45	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.69	97.29	P41212	56.4117, 56.4117, 79.9383, 90, 90, 90
cypD-860_57s	mtz002	PROC_1_EDNA_proc	ep_cypD-860_57s_w1_run1_anom_truncate.mtz	1.14	84.49	P41212	57.328, 57.328, 87.346, 90, 90, 90
cypD-862_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.89	99.95	P41212	57.402, 57.402, 88.423, 90, 90, 90
cypD-863_2min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.24	94.96	P41212	56.917, 56.917, 86.828, 90, 90, 90
cypD-864_1min	mtz003	PROC_1_XDSAPP	cypD-864_1min_w1_1_95_F_plus_F_minus_reindexed_P41212.mtz	1.36	97.99	P41212	55.835, 55.835, 80.146, 90, 90, 90
cypD-865_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.1	97.94	P41212	57.319, 57.319, 87.742, 90, 90, 90
cypD-866_6min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.05	81.21	P41212	57.231, 57.231, 87.54, 90, 90, 90
cypD-867_5min30	mtz003	PROC_1_XDSAPP	cypD-867_5min30_w1_1_92_F_plus_F_minus.mtz	1.06	95.51	P41212	57.312, 57.312, 87.716, 90, 90, 90
cypD-869_5min	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.42	97.96	P41212	56.909, 56.909, 87.2498, 90, 90, 90
cypD-872_2min30	mtz002	PROC_1_EDNA_proc	ep_cypD-872_2min30_w1_run1_anom_truncate_reindexed_P41212.mtz	2.32	95.81	P41212	55.673, 55.673, 80.398, 90, 90, 90
cypD-875_4min30	mtz002	PROC_1_XDSAPP	cypD-875_4min30_w1_1_92_F_plus_F_minus.mtz	1.42	96.72	P41212	55.502, 55.502, 85.63, 90, 90, 90

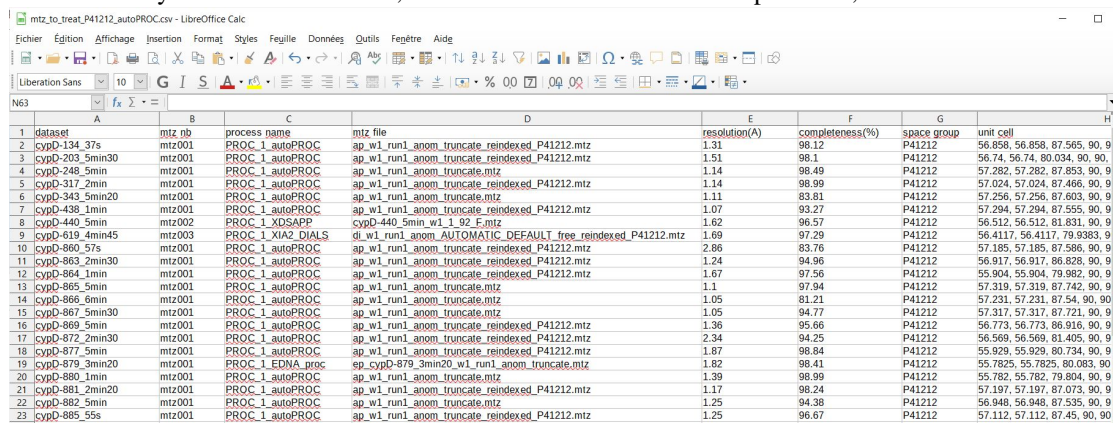
- Option example: –best 2

List only 2 first mtz, when exist, with best completeness, for each dataset.

dataset	mtz nb	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
cypD-134_37s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.31	98.12	P41212	56.858, 56.858, 87.565, 90, 90, 90
cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.51	98.1	P41212	56.74, 56.74, 80.034, 90, 90, 90
cypD-203_5min30	mtz002	PROC_1_XDSAPP	cypD-203_5min30_w1_1_89_F_reindexed_P41212.mtz	1.32	90.47	P41212	56.593, 56.593, 79.576, 90, 90, 90
cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 90, 90
cypD-248_5min	mtz003	PROC_1_XDSAPP	cypD-248_5min_w1_1_96_F_reindexed_P41212.mtz	1.03	85.68	P41212	57.274, 57.274, 87.837, 90, 90, 90
cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.14	98.99	P41212	57.024, 57.024, 87.466, 90, 90, 90
cypD-317_2min	mtz002	PROC_1_EDNA_proc	ep_cypD-317_2min_w1_run1_anom_truncate_reindexed_P41212.mtz	1.07	95.21	P41212	56.9836, 56.9836, 87.471, 90, 90, 90
cypD-343_5min20	mtz002	PROC_1_EDNA_proc	ep_cypD-343_5min20_w1_run1_anom_truncate.mtz	1.11	83.97	P41212	57.235, 57.235, 87.572, 90, 90, 90
cypD-343_5min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.11	83.81	P41212	57.256, 57.256, 87.603, 90, 90, 90
cypD-438_1min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F.mtz	1.62	96.57	P41212	56.512, 56.512, 81.831, 90, 90, 90
cypD-619_4min45	mtz003	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.69	97.29	P41212	56.4117, 56.4117, 79.9383, 90, 90, 90
cypD-619_4min45	mtz001	PROC_1_EDNA_proc	ep_cypD-619_4min45_w1_run1_anom_truncate_reindexed_P41212.mtz	1.96	91.14	P41212	55.887, 55.887, 79.878, 90, 90, 90
cypD-860_57s	mtz002	PROC_1_EDNA_proc	ep_cypD-860_57s_w1_run1_anom_truncate.mtz	1.14	84.49	P41212	57.328, 57.328, 87.346, 90, 90, 90
cypD-860_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	2.86	83.76	P41212	57.185, 57.185, 87.586, 90, 90, 90
cypD-863_2min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.24	94.96	P41212	56.917, 56.917, 86.828, 90, 90, 90
cypD-864_1min	mtz003	PROC_1_XDSAPP	cypD-864_1min_w1_1_95_F_reindexed_P41212.mtz	1.36	96.05	P41212	55.835, 55.835, 80.146, 90, 90, 90
cypD-865_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.1	97.94	P41212	57.319, 57.319, 87.742, 90, 90, 90
cypD-865_5min	mtz004	PROC_1_XIA2_DIALS	dl_w1_run1_anom_AUTOMATIC_DEFAULT_free.mtz	1.01	87.2	P41212	57.2763, 57.2763, 87.709, 90, 90, 90
cypD-866_6min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.05	81.21	P41212	57.271, 57.271, 87.74, 90, 90, 90

- Option example: `–autoproc`

List only mtz from autoPROC, or if none list mtz with best completeness, for each dataset.



dataset	mtz file	process name	mtz file	resolution(A)	completeness(%)	space group	unit cell
cypD-134_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.31	98.12	P41212	56.858, 56.858, 87.565, 90, 9
cypD-203_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.51	98.1	P41212	56.74, 56.74, 80.034, 90, 9
cypD-248_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.14	98.49	P41212	57.282, 57.282, 87.853, 90, 9
cypD-317_2min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.14	98.99	P41212	57.024, 57.024, 87.406, 90, 9
cypD-343_5min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.11	83.81	P41212	57.256, 57.256, 87.603, 90, 9
cypD-438_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.07	93.27	P41212	57.294, 57.294, 87.555, 90, 9
cypD-440_5min	mtz002	PROC_1_XDSAPP	cypD-440_5min_w1_1_92_F.mtz	1.62	96.57	P41212	56.512, 56.512, 81.831, 90, 9
cypD-619_4min45	mtz003	PROC_1_XIA2_DIALS	di_w1_run1_anom_AUTOMATIC_DEFAULT_free_reindexed_P41212.mtz	1.69	97.29	P41212	56.4117, 56.4117, 79.9383, 9
cypD-860_57s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	2.86	83.76	P41212	57.185, 57.185, 87.586, 90, 9
cypD-863_2min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.24	94.96	P41212	56.917, 56.917, 86.828, 90, 9
cypD-864_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.67	97.56	P41212	55.904, 55.904, 79.982, 90, 9
cypD-865_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.1	97.94	P41212	57.319, 57.319, 87.742, 90, 9
cypD-866_6min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.05	81.21	P41212	57.231, 57.231, 87.54, 90, 9
cypD-867_5min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.05	94.77	P41212	57.317, 57.317, 87.721, 90, 9
cypD-869_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.36	95.66	P41212	56.773, 56.773, 86.916, 90, 9
cypD-872_2min30	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	2.34	94.25	P41212	56.569, 56.569, 81.405, 90, 9
cypD-877_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.87	98.84	P41212	55.929, 55.929, 80.734, 90, 9
cypD-879_3min20	mtz001	PROC_1_EDNA_proc	sp_cypD-879_3min20_w1_run1_anom_truncate.mtz	1.82	98.41	P41212	55.7825, 55.7825, 80.083, 90
cypD-880_1min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.39	98.99	P41212	55.782, 55.782, 79.804, 90, 9
cypD-881_2min20	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.17	98.24	P41212	57.197, 57.197, 87.073, 90, 9
cypD-882_5min	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate.mtz	1.25	94.38	P41212	56.948, 56.948, 87.535, 90, 9
cypD-885_55s	mtz001	PROC_1_autoPROC	ap_w1_run1_anom_truncate_reindexed_P41212.mtz	1.25	96.67	P41212	57.112, 57.112, 87.45, 90, 9

3. List mtz files with mode and ligand information for running Phenix

Note: Phenix options for the different modes are specified *hereafter*.

For each dataset, write in a ‘launch csv’ file:

- if ligand cif file is present for search when asked
- mode that will be launched depending on mode asked, the presence (or not) of ligand cif file and data quality
- information in case mode is different from mode asked
- result folder name

Limits for poor data: There are minimum limits to process in ‘full’ or ‘allsg’ modes. These limits can be modified in config.py file (after what easypipe should be reinstalled).

- minimum completeness (default = 70%)
- minimum resolution (default = 3.75)

Poor data will be treated in ‘fast’ mode.

Option examples:

- Option example: `–mode fast` (default)

Phenix uses a simple rigid-body refinement for model placement, which is faster and most of the time sufficient if the input model is already close enough to the target structure.

launch_mtz_P41212_FAST_best1.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Libération Sans 10 G I S A

X47 \sum =

	A	B	I	J	K	L	M	N	O
1	dataset	mtz nb	mode	ligand search	CC	nb ligands	ligand	information	to treat
2	cypD-134_37s	mtz001	fast	no					yes
3	cypD-203_5min30	mtz001	fast	no					yes
4	cypD-248_5min	mtz001	fast	no					yes
5	cypD-317_2min	mtz001	fast	no					yes
6	cypD-343_5min20	mtz002	fast	no					yes
7	cypD-438_1min	mtz001	fast	no					yes
8	cypD-440_5min	mtz002	fast	no					yes
9	cypD-619_4min45	mtz003	fast	no					yes
10	cypD-860_57s	mtz004	fast	no					yes
11	cypD-862_5min	mtz001	fast	no					yes
12	cypD-863_2min30	mtz004	fast	no					yes
13	cypD-864_1min	mtz003	fast	no					yes
14	cypD-865_5min	mtz001	fast	no					yes
15	cypD-866_6min	mtz001	fast	no					yes
16	cypD-867_5min30	mtz003	fast	no					yes
17	cypD-869_5min	mtz004	fast	no					yes
18	cypD-872_2min30	mtz002	fast	no					yes
19	cypD-875_4min30	mtz002	fast	no					yes
20	cypD-877_5min	mtz003	fast	no					yes
21	cypD-879_3min20	mtz001	fast	no					yes
22	cypD-880_1min	mtz001	fast	no					yes
23	cypD-881_2min20	mtz001	fast	no					yes
24	cypD-882_5min	mtz001	fast	no					yes
25	cypD-885_55s	mtz001	fast	no					yes
26	cypD-886_5min	mtz001	fast	no					yes
27	cypD-887_4min	mtz001	fast	no					yes
28	cypD-888_1min55	mtz001	fast	no					yes
29	cypD-890_1min30	mtz004	fast	no					yes
30	cypD-895_1min	mtz004	fast	no					yes

- Option example: `--mode full`

Phenix will try rigid-body refinement first, then run Phaser if the R-free is too high (>0.4), it will run AutoBuild after initial refinement only if R-free is greater than the `max_r_free` cutoff = 0.3.

launch_mtz_P41212_FULL_best1.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Y49:Y50

	A	B	I	J	K	L	M	N	O
1	dataset	mtz nb	mode	ligand search	CC	nb ligands	ligand	information	to treat
2	cypD-134_37s	mtz001	full	no					yes
3	cypD-203_5min30	mtz001	full	no					yes
4	cypD-248_5min	mtz001	full	no					yes
5	cypD-317_2min	mtz001	full	no					yes
6	cypD-343_5min20	mtz002	full	no					yes
7	cypD-438_1min	mtz001	full	no					yes
8	cypD-440_5min	mtz002	full	no					yes
9	cypD-619_4min45	mtz003	full	no					yes
10	cypD-860_57s	mtz004	full	no					yes
11	cypD-862_5min	mtz001	full	no					yes
12	cypD-863_2min30	mtz004	full	no					yes
13	cypD-864_1min	mtz003	full	no					yes
14	cypD-865_5min	mtz001	full	no					yes
15	cypD-866_6min	mtz001	full	no					yes
16	cypD-867_5min30	mtz003	full	no					yes
17	cypD-869_5min	mtz004	full	no					yes
18	cypD-872_2min30	mtz002	full	no					yes
19	cypD-875_4min30	mtz002	full	no					yes
20	cypD-877_5min	mtz003	full	no					yes
21	cypD-879_3min20	mtz001	full	no					yes
22	cypD-880_1min	mtz001	full	no					yes
23	cypD-881_2min20	mtz001	full	no					yes
24	cypD-882_5min	mtz001	full	no					yes
25	cypD-885_55s	mtz001	full	no					yes
26	cypD-886_5min	mtz001	full	no					yes
27	cypD-887_4min	mtz001	full	no					yes
28	cypD-888_1min55	mtz001	full	no					yes
29	cypD-890_1min30	mtz004	full	no					yes
30	cypD-895_1min	mtz004	full	no					yes

- Option example: `--mode allsg`

In this mode, mtz will be treated regardless of the space group. Phenix will run Phaser, then run AutoBuild after initial refinement only if R-free is greater than the `max_r_free` cutoff = 0.3.

launch_mtz_allsg_ALLSG_best1.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

W44

	A	B	I	J	K	L	M	N	O
1	dataset	mtz_nb	mode	ligand search	CC	nb ligands	ligand	information	to treat
2	cypD-134_37s	mtz001	allsg	no					yes
3	cypD-172_1min20	mtz003	allsg	no					yes
4	cypD-203_5min30	mtz001	allsg	no					yes
5	cypD-248_5min	mtz001	allsg	no					yes
6	cypD-317_2min	mtz001	allsg	no					yes
7	cypD-343_5min20	mtz002	allsg	no					yes
8	cypD-438_1min	mtz001	allsg	no					yes
9	cypD-440_5min	mtz001	allsg	no					yes
10	cypD-619_4min45	mtz003	allsg	no					yes
11	cypD-860_57s	mtz004	allsg	no					yes
12	cypD-861_5min30	mtz001	fast	no			resolution > 3.75Å		yes
13	cypD-862_5min	mtz001	allsg	no					yes
14	cypD-863_2min30	mtz004	allsg	no					yes
15	cypD-864_1min	mtz003	allsg	no					yes
16	cypD-865_5min	mtz001	allsg	no					yes
17	cypD-866_6min	mtz001	allsg	no					yes
18	cypD-867_5min30	mtz003	allsg	no					yes
19	cypD-869_5min	mtz004	allsg	no					yes
20	cypD-872_2min30	mtz002	allsg	no					yes
21	cypD-874_5min30	mtz001	allsg	no					yes
22	cypD-875_4min30	mtz002	allsg	no					yes
23	cypD-877_5min	mtz003	allsg	no					yes
24	cypD-878_5min	mtz002	allsg	no					yes
25	cypD-879_3min20	mtz001	allsg	no					yes
26	cypD-880_1min	mtz001	allsg	no					yes
27	cypD-881_2min20	mtz001	allsg	no					yes
28	cypD-882_5min	mtz001	allsg	no					yes
29	cypD-884_5min	mtz002	allsg	no					yes
30	cypD-885_55s	mtz001	allsg	no					yes
31	cypD-886_5min	mtz001	allsg	no					yes
32	cypD-887_4min	mtz001	allsg	no					yes
33	cypD-888_1min55	mtz001	allsg	no					yes
34	cypD-889_2min	mtz001	allsg	no					yes

- Option example: `--mode full --lig`

Phenix will be run in ‘full’ mode. Then ligand will be searched with [LigandFit](https://www.phenix-online.org/documentation/reference/ligandfit.html)² and placed if cutoff model-to-map CC is more than 0.7 (default). This cutoff can be changed with ‘`--cclig`’ option. The number of ligands to be placed (default=1) can be changed with ‘`--nblig`’ option.

² <https://www.phenix-online.org/documentation/reference/ligandfit.html>

launch_mtz_P41212_FULL-LIG_best1.csv - LibreOffice Calc

Fichier Édition Affichage Insertion Format Styles Feuille Données Outils Fenêtre Aide

Libération Sans 10 G I S A

Y46 fx Σ =

	A	B	I	J	K	L	M	N	O
1	dataset	mtz_nb	mode	ligand search	CC	nb ligands	ligand	information	to treat
2	cypD-134_37s	mtz001	full	yes	0.7	1	134		yes
3	cypD-203_5min30	mtz001	full	yes	0.7	1	203		yes
4	cypD-248_5min	mtz001	full	yes	0.7	1	248		yes
5	cypD-317_2min	mtz001	full	yes	0.7	1	317		yes
6	cypD-343_5min20	mtz002	full	yes	0.7	1	343		yes
7	cypD-438_1min	mtz001	full	yes	0.7	1	438		yes
8	cypD-440_5min	mtz002	full	yes	0.7	1	440		yes
9	cypD-619_4min45	mtz003	full	yes	0.7	1	619		yes
10	cypD-860_57s	mtz004	full	yes	0.7	1	860		yes
11	cypD-862_5min	mtz001	full	yes	0.7	1	862		yes
12	cypD-863_2min30	mtz004	full	yes	0.7	1	863		yes
13	cypD-864_1min	mtz003	full	yes	0.7	1	864		yes
14	cypD-865_5min	mtz001	full	yes	0.7	1	865		yes
15	cypD-866_6min	mtz001	full	yes	0.7	1	866		yes
16	cypD-867_5min30	mtz003	full	yes	0.7	1	867		yes
17	cypD-869_5min	mtz004	full	yes	0.7	1	869		yes
18	cypD-872_2min30	mtz002	full	yes	0.7	1	872		yes
19	cypD-875_4min30	mtz002	full	yes	0.7	1	875		yes
20	cypD-877_5min	mtz003	full	yes	0.7	1	877		yes
21	cypD-879_3min20	mtz001	full	yes	0.7	1	879		yes
22	cypD-880_1min	mtz001	full	yes	0.7	1	880		yes
23	cypD-881_2min20	mtz001	full	yes	0.7	1	881		yes
24	cypD-882_5min	mtz001	full	yes	0.7	1	882		yes
25	cypD-885_55s	mtz001	full	yes	0.7	1	885		yes
26	cypD-886_5min	mtz001	full	yes	0.7	1	886		yes
27	cypD-887_4min	mtz001	full	yes	0.7	1	887		yes
28	cypD-888_1min55	mtz001	full	yes	0.7	1	888		yes
29	cypD-890_1min30	mtz004	full	yes	0.7	1	890		yes
30	cypD-895_1min	mtz004	full	yes	0.7	1	895		yes

4. Launch Phenix according to chosen mode and options

`phenix.ligand_pipeline`¹ is launched for each mtz file according to chosen mode and options, as listed in the ‘launch csv’ file (see 3. above).

If this ‘launch csv’ exists and you have modified something like adding a ligand cif for example, ‘launch’ mode should be run again, but in simulation mode so as it generates a new correct launch csv file instead of using existing one. When a new ‘launch csv’ file has been generated, just run the same command without simulation mode.

Example:

```
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --autoproc --simulate
then:
$ easypipe.py PROCESSED_DATA launch my_ref_folder --mode full --lig --autoproc
```

Simulation mode also allows to modify the ‘to treat’ column of the ‘launch csv’ file (replacing ‘yes’ by ‘no’). Useful if you want to run some options only on some mtz. Then just run the same command without simulation mode. You

can also modify the following columns: ‘mode’, ‘ligand search’, ‘CC’, ‘nb ligands’, as long as you know what you are doing.

5. Write results

At the end of each ‘launch’ subcommand, results are copied in a ‘RESULT’ folder.

In datasets folders, copy of:

- corresponding processed data and logs (useful for deposition at the PDB)
- pdb and mtz result files
- phenix cif file if ligand found
- ligand folder, if exists
- pdb of ligand(s) placed by LigandFit (all CC)

```

RESULTS_P41212/
├── cypD-134_37s
│   ├── cypD-134_37s_mtz001_CC0.7_nblig1_fast-lig.mtz
│   ├── cypD-134_37s_mtz001_CC0.7_nblig1_fast-lig.pdb
│   ├── cypD-134_37s_mtz001_CC0.7_nblig1_full-lig.mtz
│   ├── cypD-134_37s_mtz001_CC0.7_nblig1_full-lig.pdb
│   ├── cypD-134_37s_mtz001_fast.mtz
│   ├── cypD-134_37s_mtz001_fast.pdb
│   ├── cypD-134_37s_mtz001_full.mtz
│   ├── cypD-134_37s_mtz001_full.pdb
│   ├── cypD-134_37s_mtz002_fast.mtz
│   ├── cypD-134_37s_mtz002_fast.pdb
│   ├── data_mtz001
│   │   ├── ap_w1_run1_anom_autoPROC.log
│   │   ├── ap_w1_run1_anom_truncate.mtz
│   │   ├── ap_w1_run1_anom_truncate_reindexed_P41212.mtz
│   │   ├── ap_w1_run1_anom_XDS_ASCII.HKL.gz
│   │   └── xtriage_ap_w1_run1_anom_truncate.log
│   ├── data_mtz002
│   │   ├── CORRECT.LP
│   │   ├── cypD-134_37s_w1_1_F_plus_F_minus.mtz
│   │   ├── cypD-134_37s_w1_1_F_plus_F_minus_reindexed_P41212.mtz
│   │   ├── phenix_xtriage.log
│   │   ├── pointless.log
│   │   ├── XDS_ASCII.HKL
│   │   └── xtriage_cypD-134_37s_w1_1_F_plus_F_minus.log
│   └── ligand
│       ├── 134.cif
│       ├── 134.pdb
│       ├── ligand_fit_mtz001_CC0.7_nblig1_fast-lig_1_1.pdb
│       ├── ligand_fit_mtz001_CC0.7_nblig1_fast-lig_1.pdb
│       ├── ligand_fit_mtz001_CC0.7_nblig1_full-lig_1_1.pdb
│       └── ligand_fit_mtz001_CC0.7_nblig1_full-lig_1.pdb
├── cypD-172_1min20
│   ├── cypD-172_1min20_mtz001_fast.mtz
│   ├── cypD-172_1min20_mtz001_fast.pdb
│   ├── cypD-172_1min20_mtz002_fast.mtz
│   ├── cypD-172_1min20_mtz002_fast.pdb
│   ├── data_mtz001
│   │   ├── ap_w1_run1_anom_autoPROC.log
│   │   ├── ap_w1_run1_anom_truncate.mtz
│   │   ├── ap_w1_run1_anom_truncate_reindexed_P41212.mtz
│   │   ├── ap_w1_run1_anom_XDS_ASCII.HKL.gz
│   │   └── xtriage_ap_w1_run1_anom_truncate.log
│   ├── data_mtz002
│   │   ├── CORRECT.LP
│   │   ├── cypD-172_1min20_w1_1_F_plus_F_minus.mtz
│   │   ├── cypD-172_1min20_w1_1_F_plus_F_minus_reindexed_P41212.mtz
│   │   ├── phenix_xtriage.log
│   │   ├── pointless.log
│   │   ├── XDS_ASCII.HKL
│   │   └── xtriage_cypD-172_1min20_w1_1_F_plus_F_minus.log
│   └── ligand
│       ├── 172.cif
│       └── 172.pdb
├── cypD-203_5min30
│   ├── cypD-203_5min30_mtz001_CC0.7_nblig1_fast-lig.mtz
│   ├── cypD-203_5min30_mtz001_CC0.7_nblig1_fast-lig.pdb
│   ├── cypD-203_5min30_mtz001_CC0.7_nblig1_full-lig.mtz
│   ├── cypD-203_5min30_mtz001_CC0.7_nblig1_full-lig.pdb
│   ├── cypD-203_5min30_mtz001_fast.mtz
│   ├── cypD-203_5min30_mtz001_fast.pdb
│   ├── cypD-203_5min30_mtz001_full.mtz
│   ├── cypD-203_5min30_mtz001_full.pdb
│   ├── cypD-203_5min30_mtz002_fast.mtz
│   ├── cypD-203_5min30_mtz002_fast.pdb
│   ├── data_mtz001
│   │   ├── ap_w1_run1_anom_autoPROC.log
│   │   ├── ap_w1_run1_anom_truncate.mtz
│   │   ├── ap_w1_run1_anom_truncate_reindexed_P41212.mtz
│   │   ├── ap_w1_run1_anom_XDS_ASCII.HKL.gz
│   │   └── xtriage_ap_w1_run1_anom_truncate.log

```

In a ‘_mtz_treated’ folder, copy of:

- csv listing datasets without mtz file
- csv with mtz list
- csv with mtz list after reindexing
- csv with mtz list sorted according to reference space group
- all 'launch' csv files, with a counter at the end of the names in case of several launches (with handmade modifications of launch csv file for example)

```
mtz_treated/
— launch_mtz_P41212_FAST_autoPROC_1.csv
— launch_mtz_P41212_FAST_best1_1.csv
— launch_mtz_P41212_FAST_best2_1.csv
— launch_mtz_P41212_FAST-LIG_best1_1.csv
— launch_mtz_P41212_FAST-LIG_best1_2.csv
— launch_mtz_P41212_FULL_autoPROC_1.csv
— launch_mtz_P41212_FULL_best1_1.csv
— launch_mtz_P41212_FULL-LIG_9_CC0.6_autoPROC_1.csv
— launch_mtz_P41212_FULL-LIG_9_CC0.6_autoPROC_2.csv
— launch_mtz_P41212_FULL-LIG_9_CC0.6_autoPROC_3.csv
— launch_mtz_P41212_FULL-LIG_9_CC0.6_best1_1.csv
— launch_mtz_P41212_FULL-LIG_9_CC0.6_best1_2.csv
— launch_mtz_P41212_FULL-LIG_best1_1.csv
— launch_mtz_P41212_FULL-LIG_best1_2.csv
— mtz_to_treat_ALL.csv
— mtz_to_treat_ALL_reindexed.csv
— mtz_to_treat_ALL_reindexed_sorted_P41212.csv
```

For each 'launch' subcommand, a csv file is created that summarizes the corresponding results for each dataset, with information on:

- success of Phenix
- failing step (in case success = no)
- resolution
- completeness
- Rwork / Rfree
- space group
- if ligand has been placed, number of ligands found, corresponding CC

```
— RESULTS_P41212_FAST_autoPROC_1.csv
— RESULTS_P41212_FAST_best1_1.csv
— RESULTS_P41212_FAST_best2_1.csv
— RESULTS_P41212_FAST-LIG_best1_1.csv
— RESULTS_P41212_FAST-LIG_best1_2.csv
— RESULTS_P41212_FULL_autoPROC_1.csv
— RESULTS_P41212_FULL_best1_1.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_autoPROC_1.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_autoPROC_2.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_autoPROC_3.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_best1_1.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_best1_2.csv
— RESULTS_P41212_FULL-LIG_best1_1.csv
— RESULTS_P41212_FULL-LIG_best1_2.csv
```

Option example: -a -mode full -lig -nblig 9 -cclig 0.6

RESULTS_P41212_FULL-Lig_9_CC06_autoPROC_4.csv - LibreOffice Calc

FileEditViewHelpInsertionFormatToolsDataWindowToolsFormatTools

Note: To compile the results of all ‘launch’ subcommands you have run, run ‘*summary*’ subcommand

Phenix options according to modes (only for information)

phenix.ligand_pipeline¹ options are the following:

- common options:

nproc=Auto

preserve_chain_id=True: Preserves the original chain ID

refine.after_ligand.hydrogens=False: Hydrogen atoms won’t be added prior to the final refinement step (else refinement significantly slower)

prune=False: disable Prune the model after refinement to remove residues and sidechains in poor density

keep_hetatms=True: prevent Phaser from resetting HETATMs occupancies to zero

refine.after_mr.update_waters=False: don’t add/remove waters automatically

- ‘fast’ mode:

skip_xtriage=True

mr=False: rigid-body refinement will be used

quick_refine=True: which will shorten both refinement steps from 6 to 3 cycles, and disable weight optimization.

build=False

skip_ligand=True

reference_structure=‘model.pdb’: If specified, phenix.find_alt_orig_sym_mate will be applied to map the solution to the reference structure (not working when Phaser with several monomers)

- ‘full’ mode:

mr=Auto: the program will try rigid-body refinement first, then run Phaser if the R-free is too high (>0.4)

build=Auto: Run AutoBuild after initial refinement. By default, this will be done if R-free is greater than the max_r_free cutoff = 0.3

autobuild.quick=True: Run AutoBuild in quick mode. Inferior results, but a huge time-saver

quick_refine=True: which will shorten both refinement steps from 6 to 3 cycles, and disable weight optimization.

- ‘allsg’ mode:

mr=True

quick_refine=False

- if ligand search:

ligand_copies=1 (except if option -nblig >1)

keep_input_restraints=True : if the input files include pre-calculated restraints for the target ligand, eLBOW will propagate these restraints instead of generating new ones.

References

EASYPEPIPE ‘SUMMARY’

This step can be run after several runs of ‘*launch*’ subcommands.

For each ‘launch’ subcommand, a ‘RESULT’ csv file is created that summarizes the corresponding results for each dataset (*see here*). So, if you have tried several options, you will have as many ‘RESULTS’ csv files.

```
— RESULTS_P41212_FAST_autoPROC_1.csv
— RESULTS_P41212_FAST_best1_1.csv
— RESULTS_P41212_FAST_best2_1.csv
— RESULTS_P41212_FAST-LIG_best1_1.csv
— RESULTS_P41212_FAST-LIG_best1_2.csv
— RESULTS_P41212_FULL_autoPROC_1.csv
— RESULTS_P41212_FULL_best1_1.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_autoPROC_1.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_autoPROC_2.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_autoPROC_3.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_best1_1.csv
— RESULTS_P41212_FULL-LIG_9_CC0.6_best1_2.csv
— RESULTS_P41212_FULL-LIG_best1_1.csv
— RESULTS_P41212_FULL-LIG_best1_2.csv
```

Then, you probably want to compile all these results for a better view.

Usage

easypipe.py data summary [-h]

Example:

```
$ easypipe.py PROCESSED_DATA summary
```

What does it do ?

In the ‘RESULT’ folder, ‘summary’ creates a ‘SUMMARY’ csv file where all datasets results are compiled.

For each datasets, redondant results are deleted and the remaining ones are sorted according to:

- ‘dataset’
- ‘SUCCESS’
- ‘ligand search’
- ‘mode’
- ‘Ligand’ (found or not)
- ‘Completeness’

This way, the first row of each dataset should be most of the time the best treatment to consider, but is always better to have a critical eye on information like completeness or resolution to be sure ...

EASYPEPIPE ‘PANDDA’

If you have more than 40 datasets, you can try to use [PanDDA](#) ¹ which is particularly suitable to the detection of weakly bound ligands such as fragments.

This step can be run after all ‘launch’ subcommands have been executed, and aims at organizing the data processed with easYPipe in order to be able to run PanDDA.

Usage

`easypipe.py data pandda [-h]`

Example:

```
$ easypipe.py ./PROCESSED_DATA/ pandda
```

What does it do ?

For each dataset, ‘pandda’ easYPipe subcommand copies in a dataset folder suitable for PanDDA:

- pdb and mtz files generated with easYPipe,
- cif and pdb files of corresponding ligand,

¹ <https://pandda.bitbucket.io/>

```
PANDDA/
├── logs
│   └── pandda-data_easypipe_2021-02-24_195235.log
├── PANDDA_P41212
│   └── data
│       ├── cypD-134_37s_mtz001
│       │   ├── 134.cif
│       │   ├── 134.pdb
│       │   ├── cypD-134_37s_mtz001_full.mtz
│       │   └── cypD-134_37s_mtz001_full.pdb
│       ├── cypD-203_5min30_mtz001
│       │   ├── 203.cif
│       │   ├── 203.pdb
│       │   ├── cypD-203_5min30_mtz001_full.mtz
│       │   └── cypD-203_5min30_mtz001_full.pdb
│       ├── cypD-248_5min_mtz001
│       │   ├── 248.cif
│       │   ├── 248.pdb
│       │   ├── cypD-248_5min_mtz001_full.mtz
│       │   └── cypD-248_5min_mtz001_full.pdb
│       ├── cypD-317_2min_mtz001
│       │   ├── 317.cif
│       │   ├── 317.pdb
│       │   ├── cypD-317_2min_mtz001_full.mtz
│       │   └── cypD-317_2min_mtz001_full.pdb
│       ├── cypD-343_5min20_mtz002
│       │   ├── 343.cif
│       │   ├── 343.pdb
│       │   ├── cypD-343_5min20_mtz002_full.mtz
│       │   └── cypD-343_5min20_mtz002_full.pdb
│       ├── cypD-438_1min_mtz001
│       │   ├── 438.cif
│       │   ├── 438.pdb
│       │   ├── cypD-438_1min_mtz001_full.mtz
│       │   └── cypD-438_1min_mtz001_full.pdb
│       ├── cypD-440_5min_mtz002
│       │   ├── 440.cif
│       │   ├── 440.pdb
│       │   ├── cypD-440_5min_mtz002_fast.mtz
│       │   └── cypD-440_5min_mtz002_fast.pdb
│       └── cypD-619_4min45_mtz003
│           ├── 619.cif
```

References