**Preparsing.v8.py specs**

Preparsing replacement for the Java code and weird handling of some file types (h5ad, mtx, rds, …). All file types should be handled by this single script.

If the file is compressed (.zip, .bz2, .gz), it will decompress it on the go, and return the new path to parse **(this is NEW, please update your end)**, for the following parsing method.

For most uncompressed files, it should run in ~200-300ms. For compressed files or .rds files, it can take a while (from ~1s to >1mn, depending on the size).

* **Required libraries**
  + **System :** pigz, pbzip2, pixz
  + **Python :** argparse, sys, os, h5py, json, numpy, pathlib, subprocess, zipfile, shutil, tarfile, polars, rpy2

Be careful, **rpy2** is a pain. It can install, but not work.

You can try the following code to check if it’s properly installed. There should be no warning or error.

import rpy2.rinterface\_lib.callbacks

rpy2.rinterface\_lib.callbacks.consolewrite\_warnerror = lambda \*args: None # Suppress console output from R

from rpy2.robjects.packages import importr

from rpy2.robjects import r, pandas2ri

base = importr('base')

* + **R:** Matrix, Seurat
* **How to run :**

python preparse.v8.py -f toto.h5ad …

* **Input**

usage: preparse.v8.py

-f INPUT\_FILE # Only this option is mandatory

[-o OUTPUT\_FOLDER] # If -o is set, an *output.json* file is created in this folder. Else, it outputs in stdio (console)

**Note:** If the folder does not exist, it creates it

[--sel SEL] # If the file is an ARCHIVE type, then --sel tells the program to extract and preparse the selected file. It’s automatically handled in case the archive contains only matrix.mtx, barcodes.tsv and features.tsv (or genes.tsv). If there are multiple .mtx files (and associated tsv, then --sel should equals the path of the .mtx file (not the .tsv, they will be automatically detected)

[--header {true,false}] # Only used if the file is in RAW\_TEXT format (default = true)

[--col {none,first,last}] # Only used if the file is in RAW\_TEXT format (default = first)

[--delim DELIM] # Only used if the file is in RAW\_TEXT format (default = \t)

?? [--row-names ROW\_NAMES] # Only used if the file is in RAW\_TEXT format

??[--col-names COL\_NAMES] # Only used if the file is in RAW\_TEXT format

-- organism?????

[--host HOST]

* **Output**

A JSON file with ONE of these 3 formats:

1. **If it’s an ARCHIVE and --sel is NOT set**

{

"detected\_format": "ARCHIVE", # FileType is ARCHIVE

"file\_path": "Dueck2015-norms.tar", # File path of the archive (it can differ from the

original path, for e.g. if it was unzipped).

"list\_files": [ # Array containing list of files in the ARCHIVE

{

"filename": "Dueck-norms/Dueck2015-log1p-deseqnorm.txt"

},

{

"filename": "Dueck-norms/Dueck2015-log1p-deseqnorm-scaled.txt"

}, … ]

}

1. **If it’s a single file (or a triplet, in case of MTX)**

{

"warnings":[“”,””,””], # List of warnings (if empty, I don’t put it, should I?)

"detected\_format": "H5AD", # FileType is H5AD (but can be any of the authorized FileTypes (see below)

"file\_path": "/data/gardeux/test.out", # File path (it can differ from the

original path, for e.g. if it was unzipped). In case of MTX, it can be a list of 3 files

"list\_groups": [ # A list of groups (can be 1 or many), all of them having the same infos

{

"group": "/X", # Group name/ path in Hdf5 file formats. If format is RAW\_TEXT, it will be “text\_file”, if file is RDS, it will be seurat\_object for a Seurat object, or data\_frame for a data.frame

"nber\_cols": 4992, # Nb cells

"nber\_rows": 32101, # Nb genes

"is\_count": 0, # 0 or 1 if the matrix is identified as integer or float

"genes": [ "ENSMUSG00000051951", "ENSMUSG00000089699", ...], # 10 top genes

"cells": [ "AAACAACGAATAGTTC-1", "AAACAAGTATCTCCCA-1", ...], # 10 top cells

"matrix": [ # 10x10 top matrix (if possible, can be smaller)

[ 0, 0, 0, 2.3872, 0, 0, 0, 0, 8.3298, 0 ],

[ 0, 0, ...],

...

]

},

{

"group": "/raw/X",

"nber\_cols": 4992,

"nber\_rows": 32101,

"is\_count": 1,

"genes": [ "ENSMUSG00000051951", "ENSMUSG00000089699", ...],

"cells": [ "AAACAACGAATAGTTC-1", "AAACAAGTATCTCCCA-1", ...],

"matrix": [

[ 0, 0, 0, 2, 0, 0, 0, 0, 8, 0 ],

[ 0, 0, ...],

...

]

}

]

}

1. **If it’s an error**

{"displayed\_error": “error message”}

* **It can handle these formats :**
  + **H5\_10X**: .h5 file from 10x
  + **H5AD**: .h5ad file (AnnData)
  + **LOOM**: .loom file
  + **RDS**: .rds file (for now only Seurat objects, but can be easily extended)
  + **ARCHIVE**: .zip or .tar with multiple files inside
  + **RAW\_TEXT**: any text file (.tsv, .csv, …)