**Instruction of removing gaps, clouds, and shadows in one target image using multiple images as input**

Yue Sun ([sunyuemy@gmail.com](mailto:sunyuemy@gmail.com)), Xiaolin Zhu (zhuxiaolin55@gmail.com)

Department of Land Surveying and Geo-Informatics  
The Hong Kong Polytechnic University

**Reference**

**Please cite the following references when using the code:**

1)Chen, J., Zhu, X. Vogelmann, J. E. Gao, F. and Jin, S. A simple and effective method for filling gaps in Landsat ETM+ SLC-off images. Remote Sensing of Environment, 2011, 115, 1053-1064

2)Zhu, X., Gao, F., Liu, D. and Chen, J. A modified neighborhood similar pixel interpolator approach for removing thick clouds in Landsat images. IEEE Geoscience and Remote Sensing Letters, 2012, 9(3), 521-525

3) Zhu, X., Liu, D. and Chen, J. A new geostatistical approach for filling gaps in Landsat ETM+ SLC-off images, Remote Sensing of Environment, 2012, 124, 49-60.

4)Zhu, X., Helmer, E.H., Chen, J., and Liu, D. An automatic system for reconstructing High-quality seasonal Landsat time series, in Remote Sensing Time Series Image Processing, CRC Press,2018

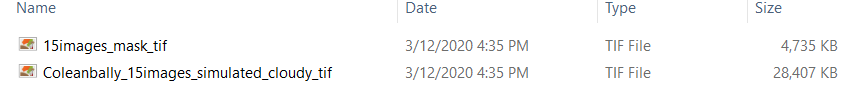
Download link:<https://data.fs.usda.gov/research/pubs/iitf/bc_iitf_2018_zhu001.pdf>

**Input data preparation**

**Two data sets needed:**

1. a time-series: all images should be stacked together，and the order is the DOY of each image. The image to be interpolated (the target image) is one of these images. the clear parts of other images should cover all contaminated pixels in the target image
2. A mask file: each layer corresponding to one image. The code should be : clear 0, contaminated pixels>0 (e.g., gap 1, shadow 2, cloud 3, or all contaminated =1), background=10

**Test data sets:**



1. A time-series with simulated missing pixels: **Coleanbally\_15images\_simulated\_cloudy\_tif.tif**, a time series stack with 15 images
2. A mask file: **15images\_mask\_tif.tif**, a mask of cloud, cloud shadow, and gap of all 15 images in the time series
3. An original time series without missing pixels: can be used to assess the accuracy of interpolation (not provided with code, please contact [zhuxiaolin55@gmail.com](mailto:zhuxiaolin55@gmail.com) to get the data).

**NOTE:**

* **The format of the output data depends on the format of the input data. If you select the GEOTIFF images as the input data, the output data is also in the GEOTIFF format.**

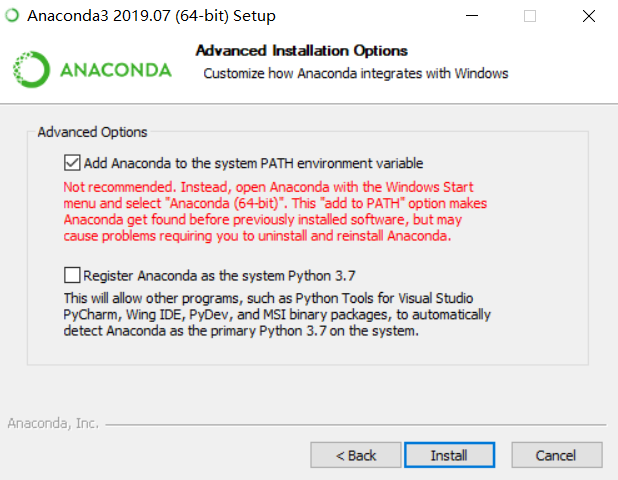
You can select any one as a target image

**Operation Steps of NSPI Time-series in Python Code**

**Step1: Configure Anaconda and PyCharm**

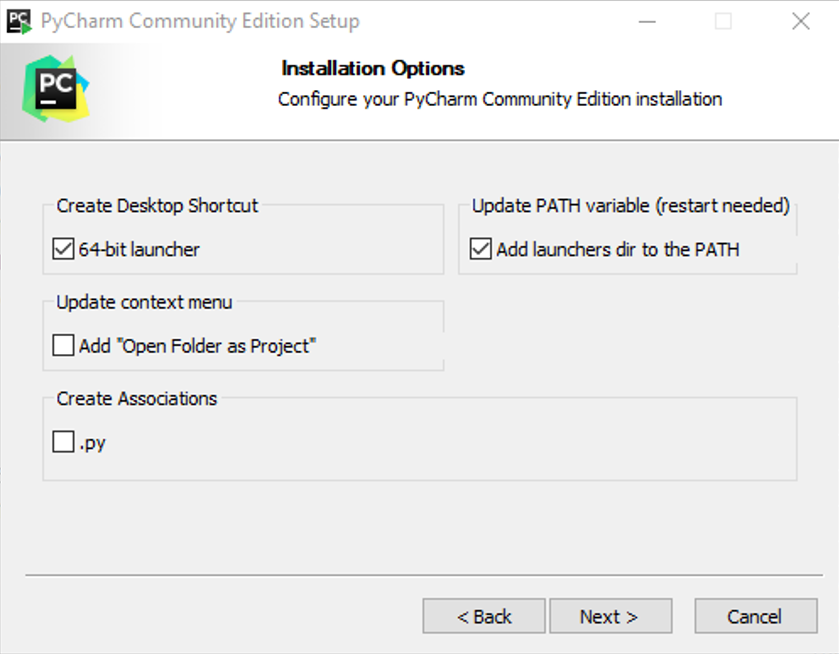
**Anaconda is selected as a virtual environment**

1. Download Anaconda from <https://www.anaconda.com/distribution/> for your operating system and install (Please check the box of “Add Anaconda to the system PATH environment variable”)



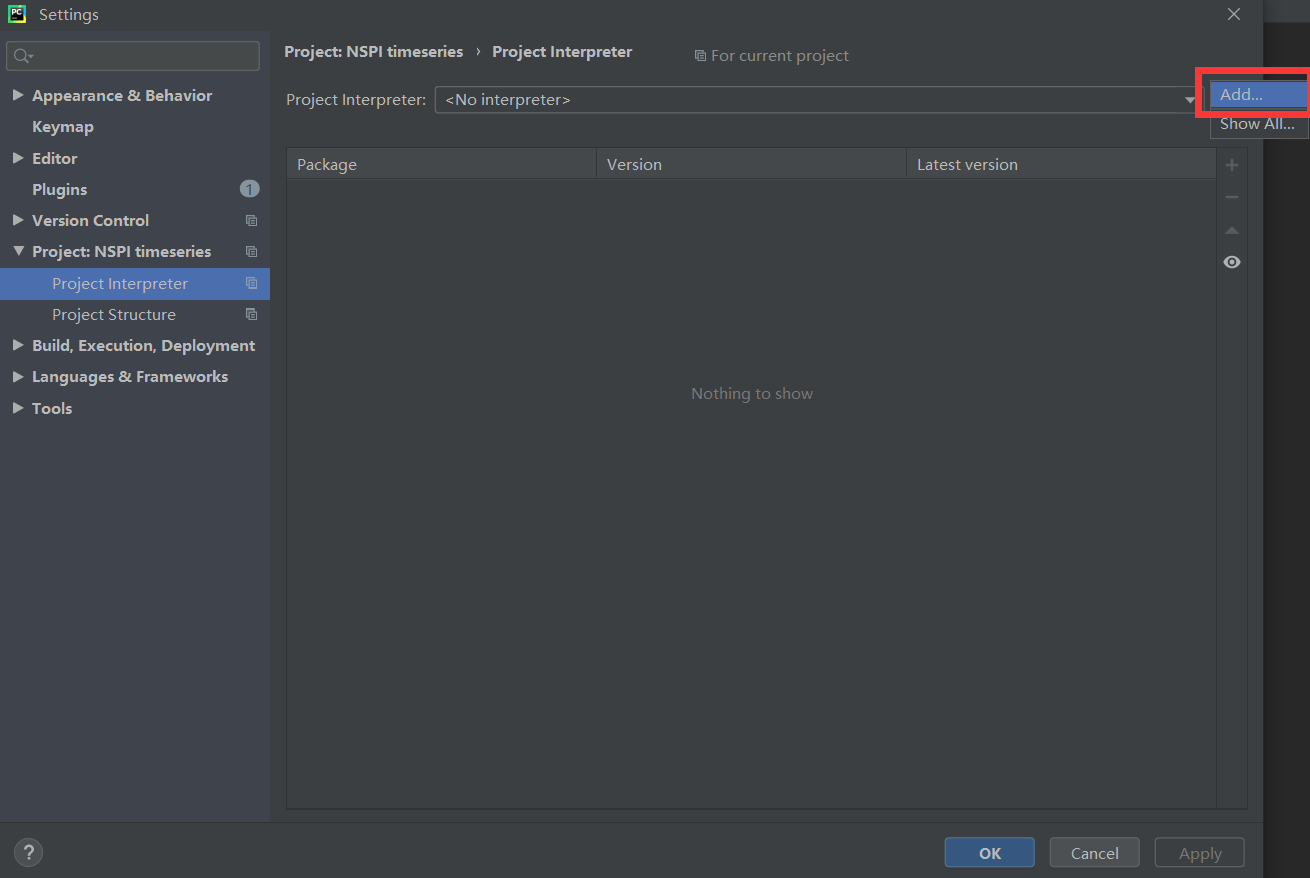
**PyCharm**

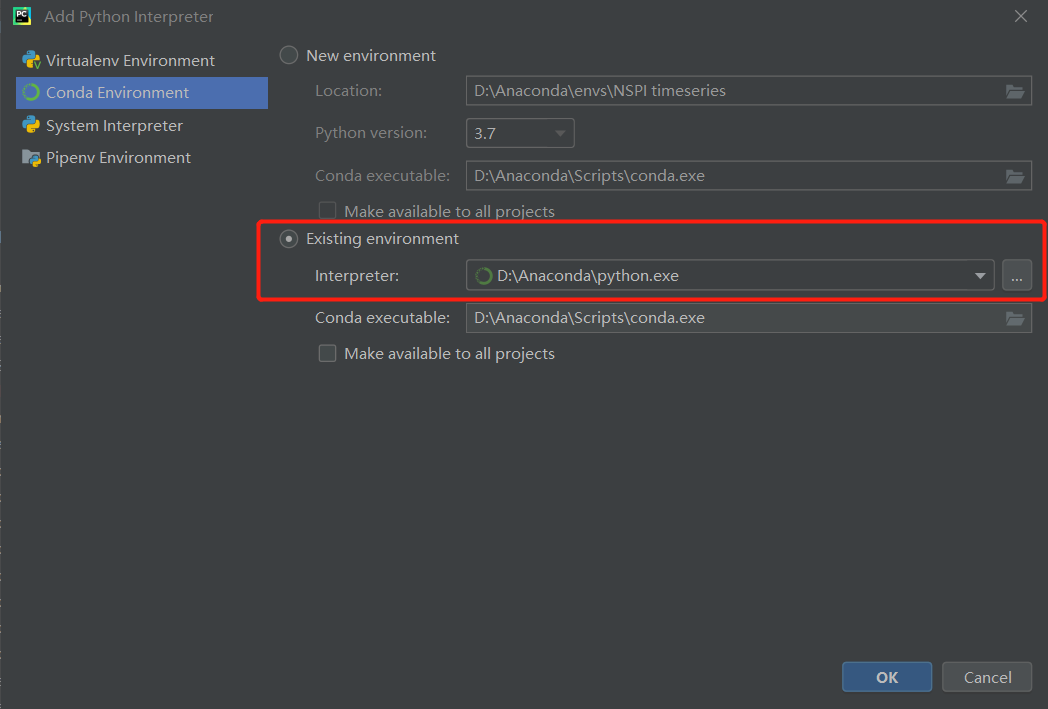
1. PyCharm Community Edition is used as an integrated development environment (IDE) for development and debugging. Download the latest version of PyCharm from <https://www.jetbrains.com/pycharm/download/#section=windows>. The free Community Edition is recommended.
2. Install PyCharm (Please check the box of “Add launchers dir to the PATH”)



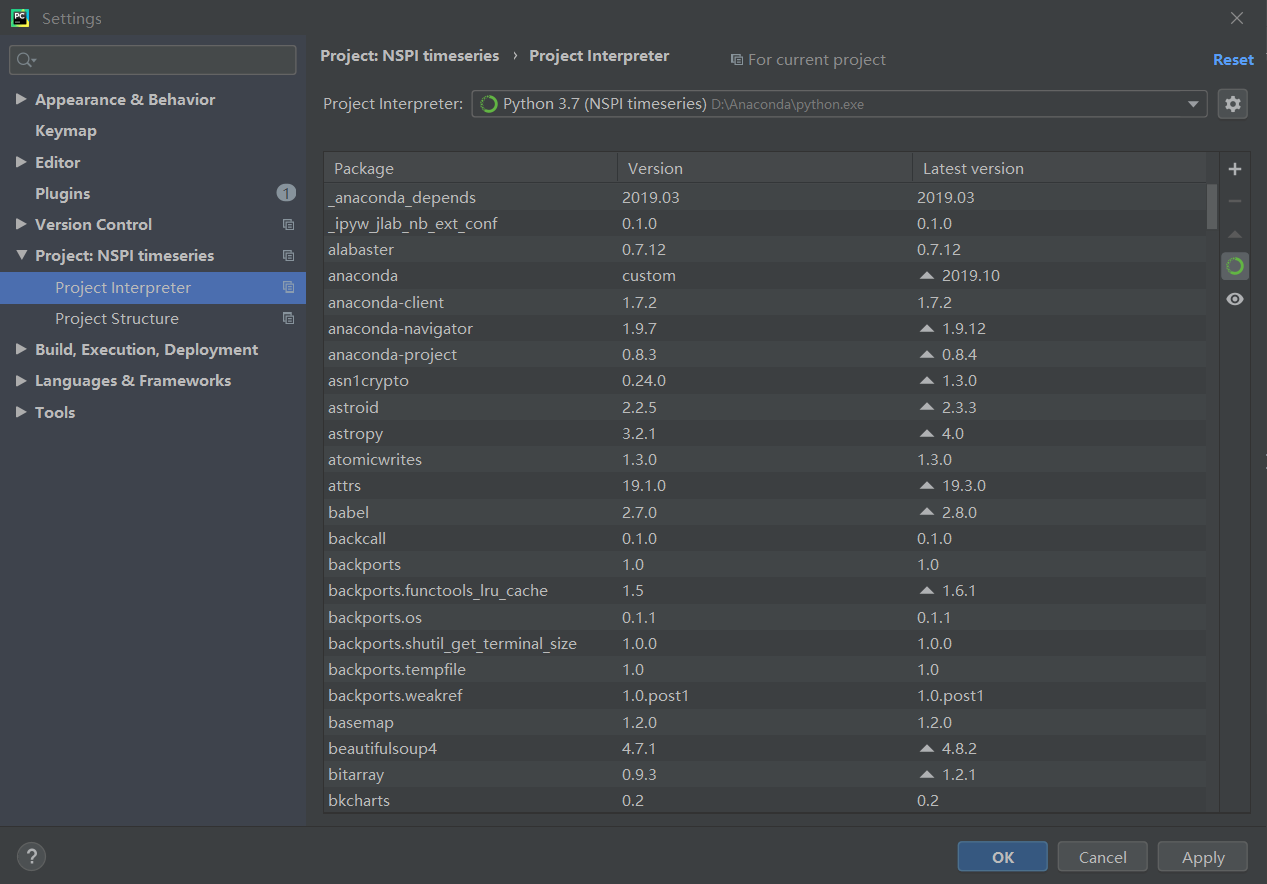
**Configure PyCharm with Anaconda**

1. Launch JetBrains PyCharm and click “File -> Open” to Open NSPI time series project (i.e. select the folder “NSPI time series” and then click “OK”)
2. Select File -> Settings -> Project -> Project Interpreter
3. Click a settings button  for adding the python.exe interpreter that you just installed in Conda Environment to the “Existing environment”. In my case, the path is “D:\Anaconda\python.exe”, then click “OK”.



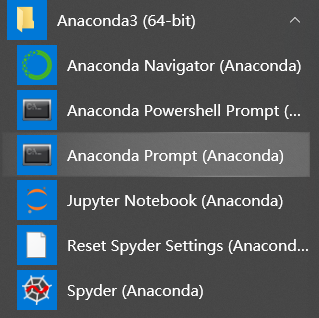


Then click “OK” in the following interface window:

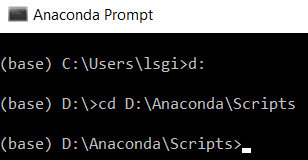


**Install the required packages**

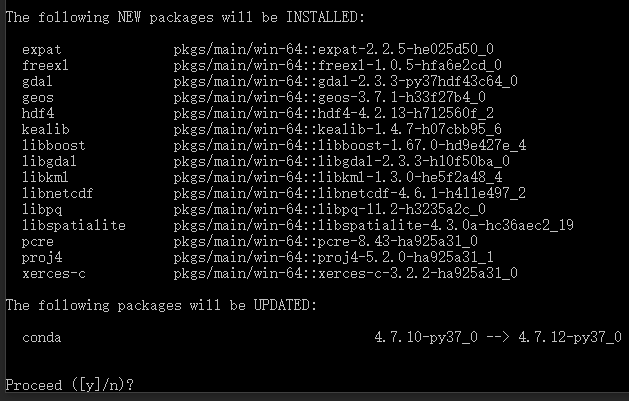
1. Open **Anaconda Prompt** in the window starter



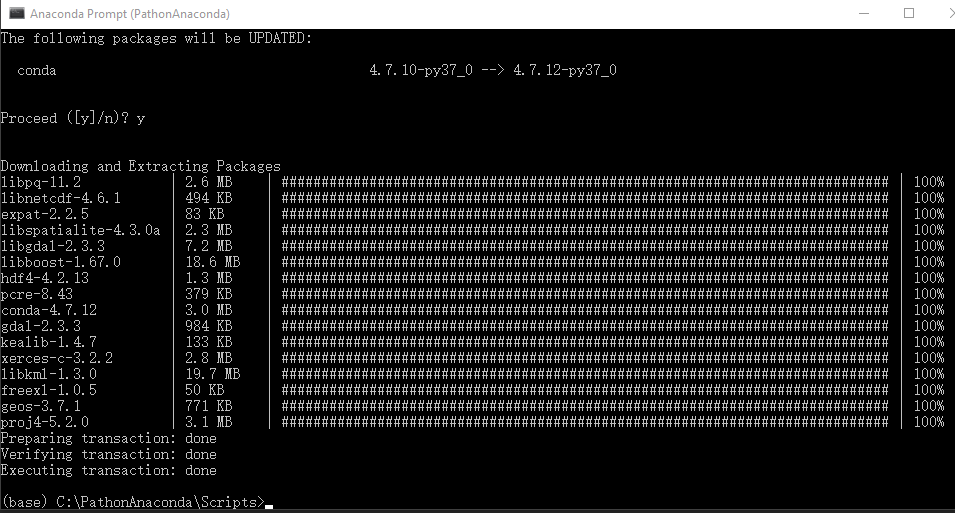
1. Locate your Python **Scripts** path (can be found within the Anaconda installation folder) and press **Enter**. In my case, the Python Scripts path is: D:\Anaconda\Scripts, so in the pop-out window, type “cd D:\Anaconda\Scripts” and the press “enter”



1. Type “**conda install gdal” and press “enter”** toinstall the required Python package **gdal** (other required packages named os, numpy, yaml, sklearn, tkinter have been included in Anaconda). Type “y” and press “enter” when the window shows the following message:



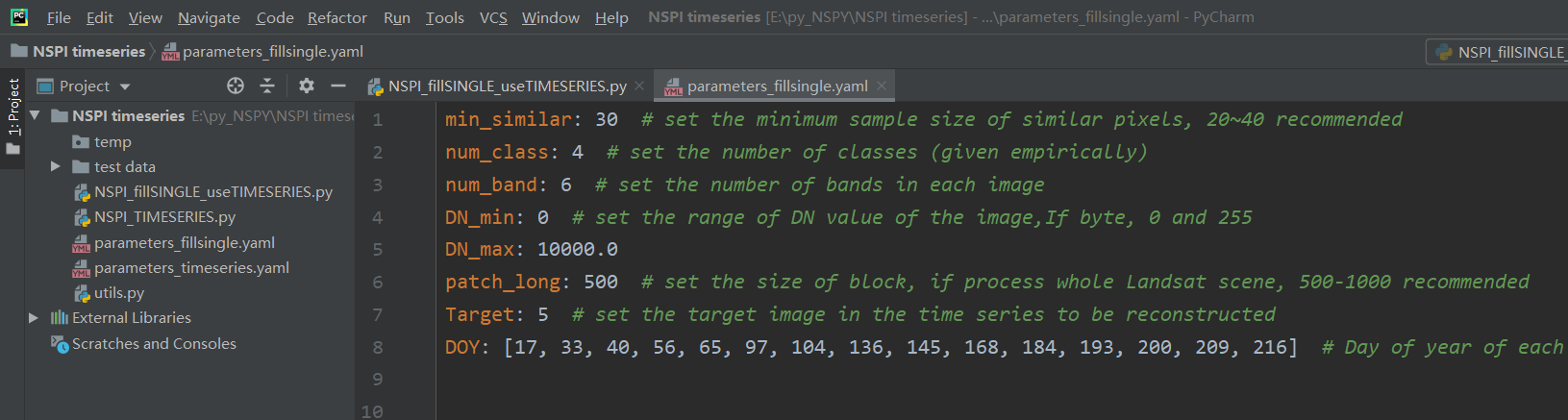
1. It is done **when you see the following text** (if no errors appear, then the package was successfully installed)



**Step 2: Set parameters and path**

1. The parameters can be set in **parameters\_fillsingle.yaml**. The default parameters in parameters\_fillsingle.yaml are used for test data, you can change these parameters according to your own data before open this parameter file.

* Follow the instruction to set parameters.
* Target is the number of the image to be interpolated
* Patch\_long is the size of block. To make the code able to run large images, we divide the images to small blocks and process each block independently. The final result is a mosaic of all blocks.
* Patch\_long should be larger than the size of cloud patches. Otherwise, it cannot find clear pixels to interpolate cloudy pixels in each block.
* We do not recommend interpolating large cloud patches because the accuracy is normally low.

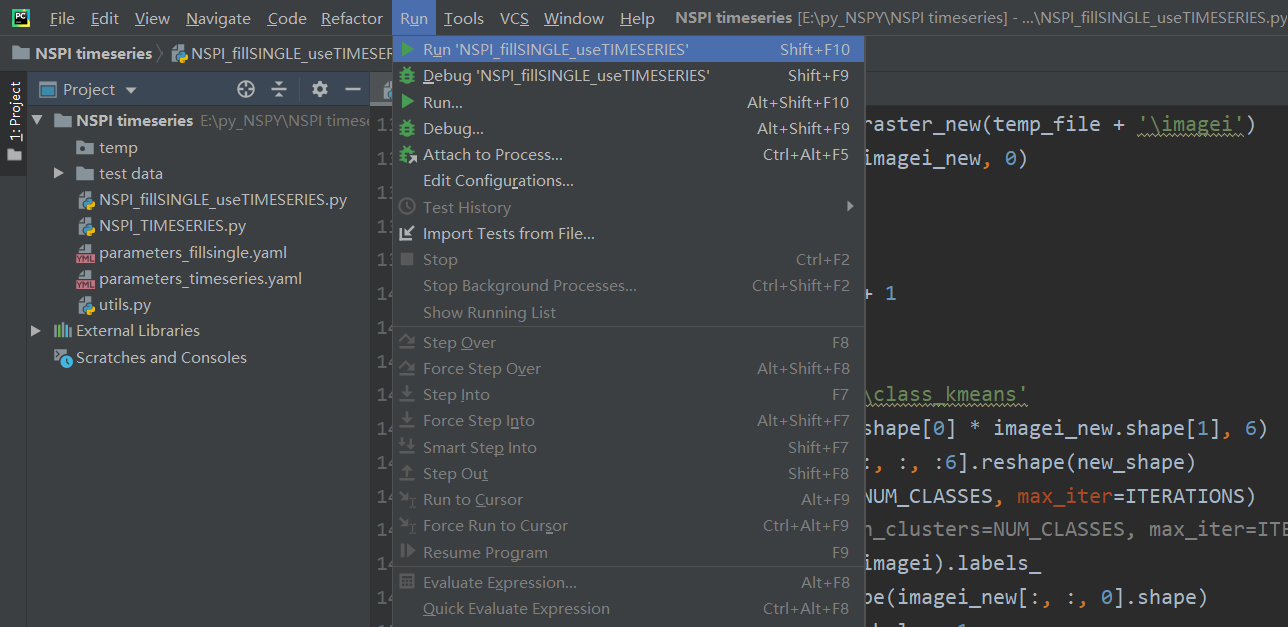


**NOTE:**

* **We do not recommend filling large clouds, e.g., size>500 pixels, since the prediction is not reliable!**

**Step 3: Run**

Click Run -> Run ‘NSPI\_fillSINGLE\_useTIMESERIES’



Open files according to the name of pop-up windows

* Open the parameter settings file (**parameters\_fillsingle.yaml**)
* Set the temporary folder
* Open the SLC-off ETM+ time-series image
* Open the mask image

The results are outputted in the same data folder

**Two results:**

1. Image after repairing: original file name with a suffix “image#\_filled\_NSPI” (# is image number in the time series)
2. Flag file to show quality: two digits AB

* A indicates which image used to fill the missing value, 1 means the first closest image used
* B indicates whether enough similar pixels used: 1 enough, 2 not enough, 3 no similar pixel found
* Results with both smaller A and B mean higher quality

**NOTE:**

* **The format of the output data depends on the format of the input data. If you select the GEOTIFF images as the input data, the output data is also in the GEOTIFF format.**

Thanks for using NSPI\_fillSINGLE\_useTIMESERIES.pro algorithm. If you meet any problems, please feel free to contact Miss Yue Sun or Dr. Xiaolin Zhu.