

LANDSAT AUTOMATED SCENE SELECTION TOOLBOX
(LASST)

By

TODD K. EASTMAN

MASTER OF SCIENCE
GEOGRAPHIC INFORMATION SYSTEMS TECHNOLOGY
FINAL PROJECT

THE UNIVERSITY OF ARIZONA

2024

To my parents who encouraged me to do my best and to learn more about our planet.

ACKNOWLEDGEMENTS

I would like to thank my professor, Yoganand Korgaonkar for his encouragement with this research and Fernando Sanchez for sparking my interest in Raster based analysis. Without their support all of this would not have been possible. I would also like to thank Gloria Fenner for volunteering to be my editor.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGEMENTS	3
LIST OF FIGURES.....	5
LIST OF ABBREVIATIONS.....	6
ABSTRACT	7
ETHICS STATEMENT	8
INTRODUCTION.....	10
METHODS	13
Workflow.....	13
Preparing the Python Environment.....	13
LASST Toolbox Process Flow.....	15
Testing.....	19
RESULTS.....	20
Point Feature Input Results	20
Line Feature Input Results.....	21
Polygon Feature Input Results.....	22
CONCLUSION	24
REFERENCES.....	26
PYTHON TOOLBOX CODE.....	27

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
Figure 1. History of Landsat missions. (USGS, 2024).....	10
Figure 2. Running Python Command Prompt.....	14
Figure 3. Selecting Python Cloned Environment in ArcGIS Pro.....	15
Figure 4. Creating a new Python Toolbox.....	16
Figure 5. LASST Input/Output Flow Diagram.....	17
Figure 6. Salt Lake City Mosaic Dataset with 50km Buffer.....	21
Figure 7. Colorado River Mosaic Dataset with 50km Buffer.....	22
Figure 8. State of Nevada Mosaic Dataset with 50km Buffer.....	23

LIST OF ABBREVIATIONS

EROS	Earth Resources Observation Satellite
ESRI	Environmental System Research Institute
GIS	Geographic Information Systems
LASST	Landsat Automated Scene Selection Toolbox
NBR	Normalized Burn Ratio
TOA	Top of Atmosphere
USGS	United States Geological Survey

ABSTRACT

The manual web-based download and use of United States Geological Survey (USGS) Landsat imagery to perform raster-based geoprocessing analysis is often time-consuming and repetitive in nature. Searching, sorting, and downloading images to cover an area of interest is overly complicated and error-prone due to the naming conventions of the output files and the excellent level of data management skills required. The purposes of this project are to automate web-based searches based upon date and coverage area and the download of Landsat imagery so that the Geographic Information Systems (GIS) analyst can focus on geoprocessing analysis and spend less time on data management. The solution to this problem is in the form of an ArcGIS toolbox developed and written in Python. Inputs to the toolbox include a selected feature (point, line, or polygon), date of interest, percent maximum cloud coverage, and a buffer length used to calculate an extent that buffers the input feature. Given the user's input, the toolbox searches the online Landsat USGS database for the date and study area of interest to locate the necessary Landsat imagery. The toolbox output is the downloaded Landsat imagery in the form of an ArcGIS mosaic dataset that encompasses the user's input extent. The toolbox typically completes this automated process within three to five minutes as compared to a manual process that may take hours if not days, depending on the input extent complexity. This toolbox will provide time-saving benefits to any analyst interested in utilizing Landsat imagery in their geoprocessing analysis.

Keywords: USGS, Landsat, ArcGIS, Python Toolbox, Automation

ETHICS STATEMENT

Geographic Information Systems (GIS) analysis encompasses a powerful set of tools and processes that can have major impacts both good and bad on individuals and our society. It is for these reasons that the results of using such tools and publishing data should be considered from an ethical point of view as to their unforeseen consequences at large. Due diligence to make careful use of tools and skills is paramount and it is the responsibility of the GIS analyst to provide the best work possible while not being swayed or influenced by the demands of others. While it is difficult to anticipate all uses of data or results, considering how this information might be used is a good place to start. Can this data be used to mistakenly sway public opinion of individuals, communities, states, or countries? Making the data and findings publicly available is a good practice to mitigate these potential exposures so that others can independently verify results and provide their own assessments of the stated information. By making results and input data as widely available as possible, one can address skepticism by broadening independent involvement to best serve the communities. By knocking down the costs associated with performing GIS-based analysis we can provide a more level playing field for all. To these concerns it is noteworthy to mention that Landsat data archives have not always been freely available to the public.

Landsat data was being sold at the international level for many years after the data was first available from the newly launched series of Landsat satellites starting in 1972. As an example, sales of Landsat data totaled around 3.5 million dollars in 1979 and increased by 22 percent to around 4.2 million dollars in 1980 (USGS, 2023). Most of these purchases were from foreign governments and academia and extraordinarily

little of this data made its way into the hands of private individuals. This policy for paying for Landsat data lasted for many years. It was not until April of 2008 was it first announced that the Landsat data archives would be made freely available to the public and became known as “Imagery for Everyone” (USGS, 2008). By December 2008, most of the newer Landsat 5 data was made publicly available, with some of the older Landsat 4 and prior data products not available until February 2009. Once all Landsat data was made publicly available, the option to purchase Landsat data was discontinued. Advancements in computer technology and processes and the forethought by our public leaders helped shape the perfect environment for the Landsat data to be made free to everyone. The price per scene of Landsat data went from a peak of nearly \$3500 per scene in the 1980’s to less than \$10 per scene once the data was made freely available (NASA, 1981). The downloads of Landsat scenes went from 86,000 scenes in 2008 to over one million scenes by August 2009 (Team, 2017). Making the data free to access provides individuals the opportunities to verify each other’s results thus providing a level of transparency. It is these policies and practices that have helped form the basis for a broad range of independent analysis to study our planet at large by a broad range of disciplines and individuals. The Landsat program has led the way in providing free and publicly available information about our planet and has fostered an unprecedented number of studies by scientists, engineers, and application GIS analysts alike. May the Landsat data product continue to be made freely available to provide the opportunities for further independent studies of our planet so there is no preferential or biased treatment of those who can afford to pay for such data products, but that all individuals have access and can contribute equally.

INTRODUCTION

Landsat satellite data collection revolutionized the analysis of Earth on a grand scale. Launched in 1972, the Landsat satellite data collection offers the longest continuous satellite-based observations of Earth. For over 50 years this data product has been made available to researchers and scientists alike, making it one of the most sought-after data sets for long-term Earth land use analysis. The project was first announced in September 1966 by Secretary of the Interior Stewart Lee Udall as part of the Earth Resources Observation Satellite (EROS) project, with the intention to make this data available to all (Department of the Interior, 1966).

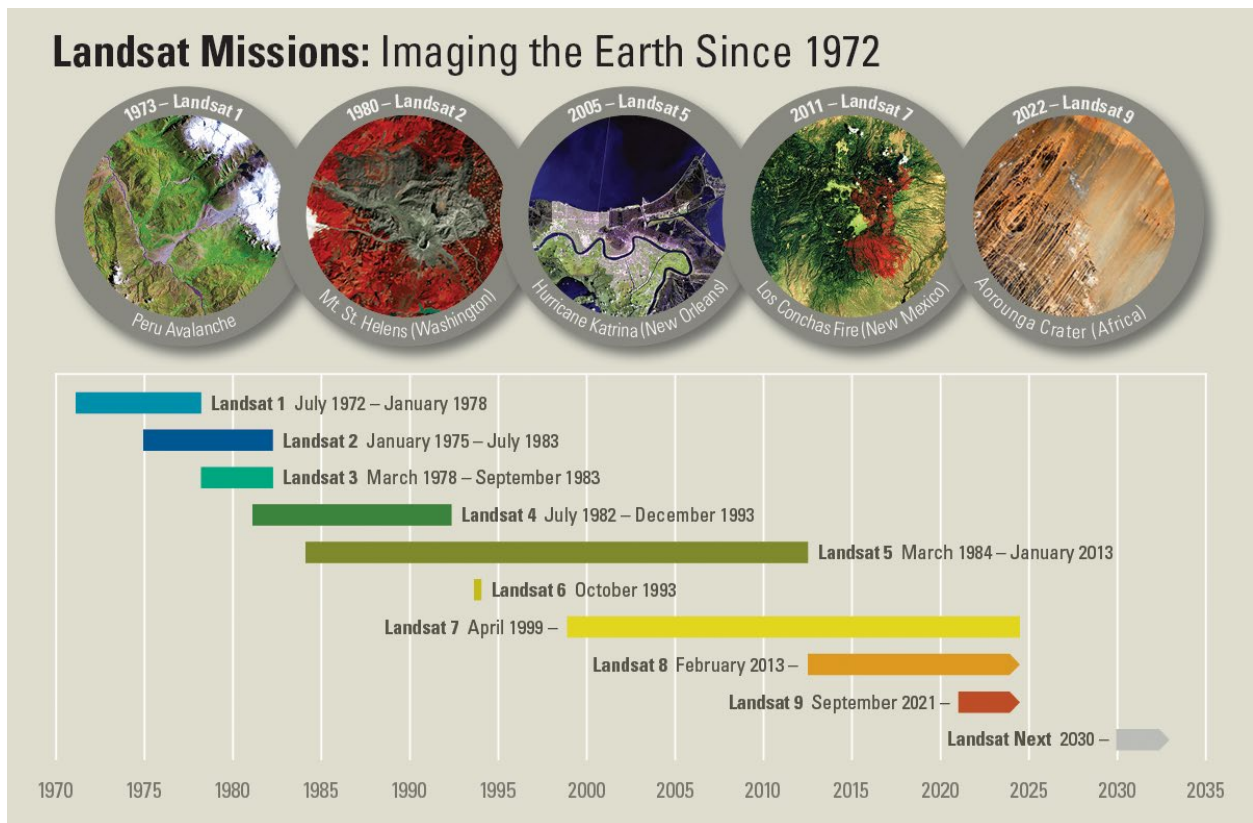


Figure 1. History of Landsat missions. (USGS, 2024)

The United States Geological Survey (USGS) has been providing the Landsat data for free online since 2008, as part of what are known as Collections. Collection one was retired in December 2022 and offered little preprocessing of the original source data. Access to Landsat archives via the USGS website is now provided in what is known as part of Collection 2. The Collection 2 formats offer a superior set of processed information to the end user and include all Landsat satellite measurements to date. This Collection 2 data is offered through many web-based portals, but most specifically through the USGS web-based EarthExplorer site (EarthExplorer, 2024). Through this site, users can input a coordinate and date ranges and EarthExplorer will provide a webpage of available Landsat scene files available to download.

Sorting through the scenes, downloading them and subsequent file organization can prove to be time-consuming. In fact, these manual operations served as the prime motivation for this project. There should be an easier way for users to utilize the Landsat archives without having to go through a manual process of inputs, sorting, and download. The solution provided for this was to write a Python based toolbox that can be used via ArcGIS Pro to make it easier for the end user. The search to see if such a toolbox existed led to a tool named landsatxplore (Forget, 2021). The landsatxplore tool provides a useful interface to the USGS Landsat data, but the interface to the end user remains complicated than what would be most useful. Given an Earth WGS-84 coordinate and data range, the landsatxplore tool provides available Landsat scenes. This would require several input coordinates and calls to the tool until the area of interest was bound by available scenes. I was looking for something a bit more user

friendly. To this I wrote a toolbox that I call Landsat Automated Scene Selection Toolbox (LASST).

The goal of this project is to provide an automated solution to the GIS analyst who wants to use date-specific Landsat scenes for a geospatial analysis. The solution is to replace the time-intensive search and download of Landsat scenes made available through the USGS EarthExplorer interface. This project fills this gap to help automate the process from start to finish.

METHODS

Workflow

LASST was developed using Microsoft Windows 11 and used the Environmental System Research Institute (ESRI) ArcGIS Pro 3.2.2 program and its support of user-developed toolboxes written in the Python programming language. ArcGIS Pro was selected based upon the author's familiarity with the tool and its wide use and acceptance within the GIS community. To accomplish this goal, a new toolbox within ArcGIS Pro was created and named LASST within a new project. To be able to run the LASST toolbox on a user-selected feature within ArcGIS Pro requires an initial one-time setup. To provide the user interface to the Landsat online database, the landsatxplore (Forget, 2021) tool was downloaded from GitHub. To download scenes from the Landsat online database, the user is required to register online through the EROS (EROS, 2024) portal as subsequent queries require a user account and password.

Preparing the Python Environment

To install and run third party python tools, the user must first clone the ArcGIS python environment (ESRI, 2023). Cloning the python environment prevents corruption of the base python installation environment which could hinder ArcGIS Pro from running properly for all projects. To clone the python environment, one must run the Python Command Prompt application as an administrator by right clicking on the application and choosing "Run as administrator" as shown in Figure 2. Steps for cloning the Python environment and installation of landsatxplore follow.

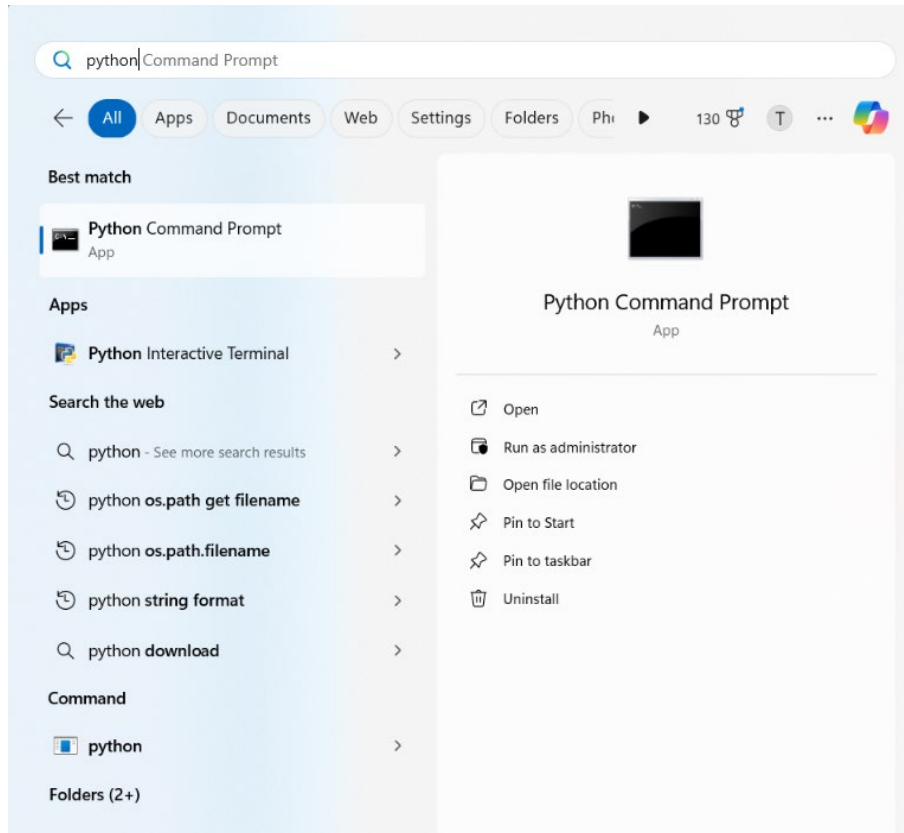


Figure 2. Running Python Command Prompt

At the Python Command Prompt search for available environments by typing

```
conda env list
```

Identify the name of the Python environment to clone and then create a new clone with the following command. Remember the name of your newly cloned environment for later selection within ArcGIS Pro.

```
conda create --clone <environment to clone> --name <new environment name> --pinned
```

At this time, the landsatxplore tool can be installed as a pip install within the Python Command Prompt with the following command.

```
pip install landsatxplore
```

Now that the newly cloned Python environment is available, select the new python environment within your ArcGIS Pro project. From ArcGIS Pro select Project->Package Manager, then select your newly cloned Python environment under the Active Environment pull-down as shown in Figure 3. Once the clone environment has been selected you can return to your project map page and save your project. At this point, the ArcGIS Pro environment is in a state to develop/utilize the LASST toolbox.

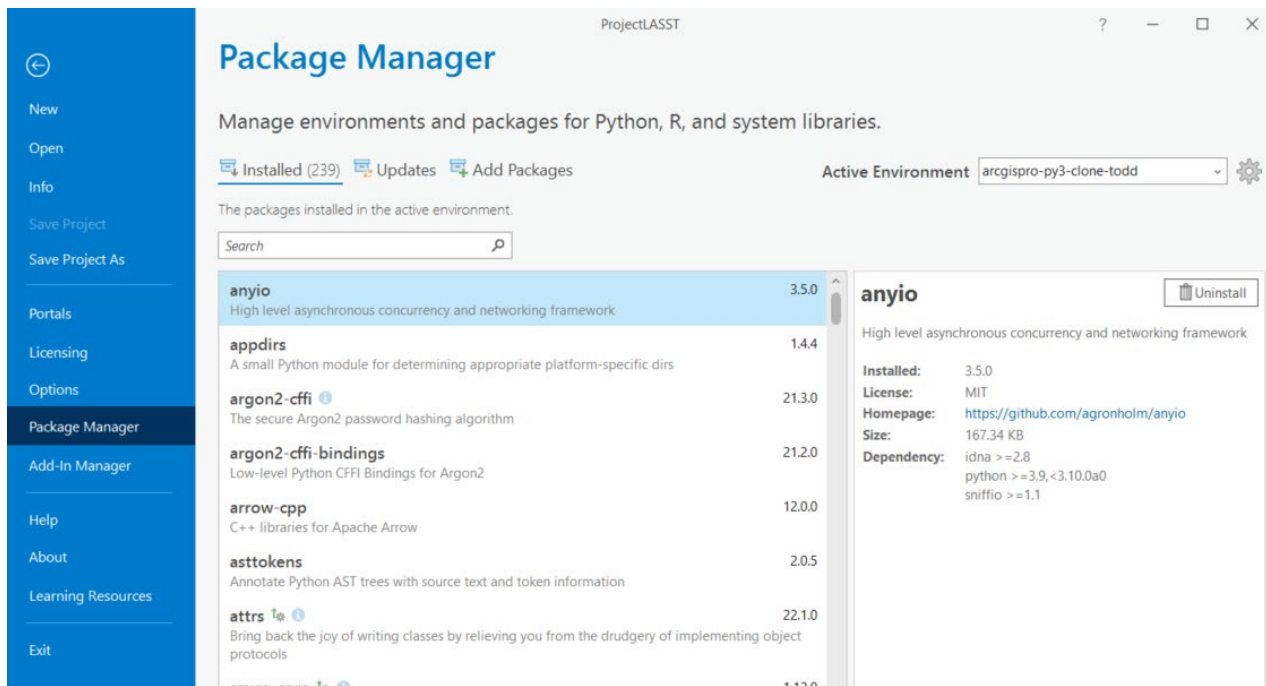


Figure 3. Selecting Python Cloned Environment in ArcGIS Pro

LASST Toolbox Process Flow

The LASST toolbox was created within ArcGIS Pro from the Catalog pane under Toolboxes by creating a New Python Toolbox as shown in Figure 4. ArcGIS creates a

template class full of input/output methods from which the developer fills in the details to achieve their goals.

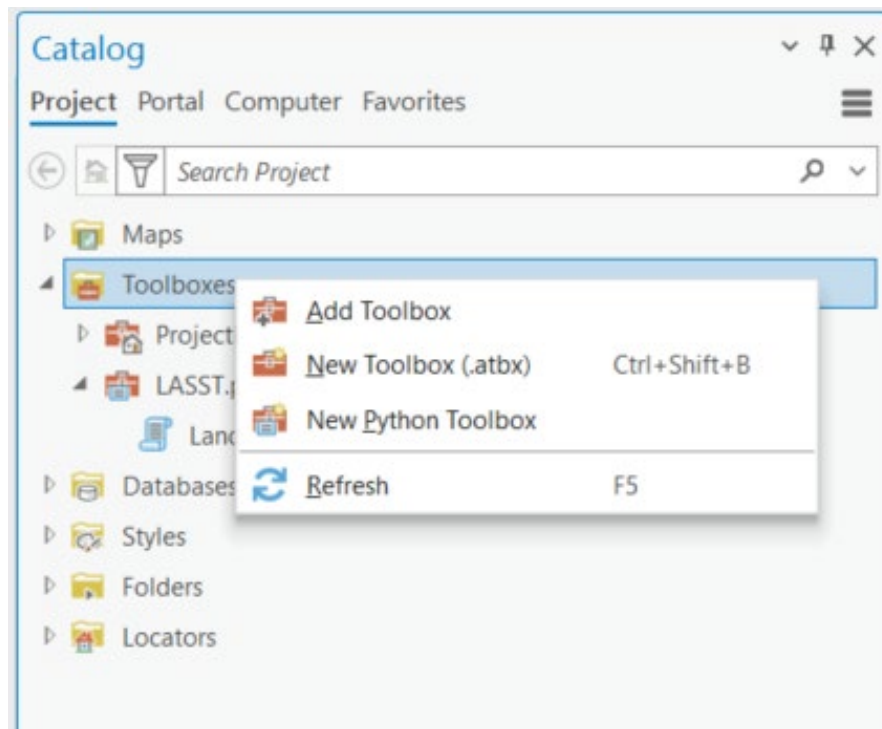


Figure 4. Creating a new Python Toolbox

The toolbox was then modified to take the following inputs as shown in Figure 5. Desired Landsat scene date in Year/Month/Day format. This is the date for which the Landsat database will be searched for available scenes. Since scenes are never captured exactly at the date of interest the toolbox takes a second date-related input parameter with one of three available values (NEAREST, BEFORE, or AFTER). When selecting the NEAREST default option, the toolbox will find an available scene that is NEAREST to the user's input date whether the available scene occurs before or after the date selected. If the user chooses BEFORE the scene nearest in time to the requested date, but only occurring at or before the requested date, will be returned.

Subsequently, when selecting AFTER, only dates at or after the input date will be returned. These modes provide the analyst with an important feature. When performing studies, there is often an event date for which scenes only BEFORE or AFTER the date of interest is required. For example, for a forest fire there is a date that the fire starts, and the user may want the nearest available scene that is just before the fire started to obtain the pre-fire scene/condition. The Max Cloud Cover is an input in percentage that will omit any scenes from the list with percent cloud cover greater than the user input. A typical value for this input would be 10, which represents scenes returned with no more than ten percent cloud cover.

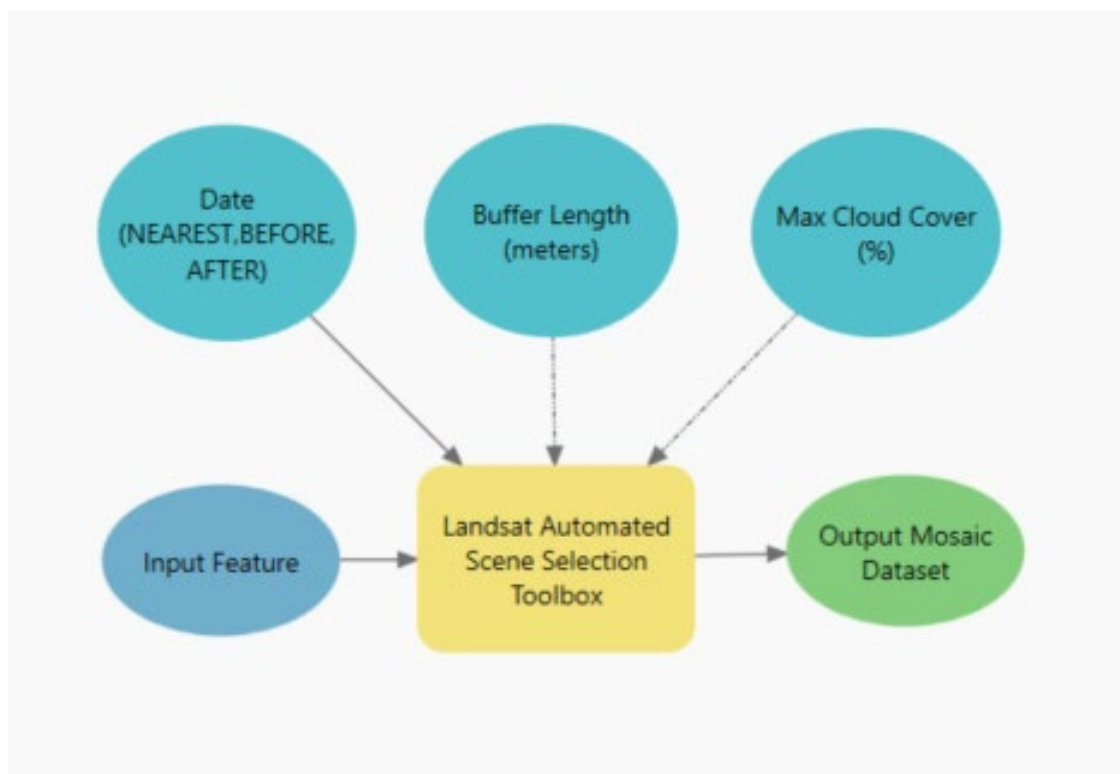


Figure 5. LASST Input/Output Flow Diagram

The Buffer Length input buffers the input feature by a length buffer equal to the Buffer Length input. This is a valuable option for point features when the user would like to analyze Landsat imagery within a certain distance surrounding a point. For example, to analyze imagery around a certain city within a 10,000-meter radius. This Buffer Length input feature is also valuable for line features such as rivers, where interest may be coverage area within a certain distance of the river. For all features, the toolbox takes the input buffer length and input feature and creates a new feature to be covered by Landsat scenes. The tool next takes the buffered input feature and overlays a fishnet grid of points. The fishnet grid of points is converted to WGS-84 regardless of the input feature spatial reference system. WGS-84 is the required coordinate system by the landsatxplore tool. These coordinates are provided to the landsatxplore tool to search for available scenes without downloading the scenes themselves. The scene meta data is returned, allowing for faster sorting and geospatial processing of available scenes. As a scene is identified it is overlaid on the fishnet grid of points. Since the Landsat scenes encompass multiple fishnet grid points, those points covered by a chosen scene are not fed back into the landsatxplore tool. This removes redundant grid points from being searched online and speeds up the feature coverage process.

Once the actual list of scenes is determined, the toolbox downloads them using parallel processing into a folder local to the project. These scenes are downloaded as compressed tar files. The parallel processing downloads and extracts each scene into its own designated scene folder within the project. This is helpful for analysis where the user may want to change one or more of the tool input parameters and scene data can be reused without repeating the download process for all scenes. The parallel process

also creates a Raster Layer for each scene as the scenes are comprised of multiple measurement bands. Once all scenes are downloaded, extracted, and multi-band layers created, the toolbox then creates an empty Mosaic Dataset and adds the layers one by one to the Mosaic Dataset. To aid the user in visualizing the multi-raster mosaic dataset, the toolbox defines and updates the Overviews/Boundaries of the output Mosaic Dataset and adds this new output feature to the user's current map selection.

Testing

The toolbox was evaluated with a varied set of input features to ensure that it could process them correctly. Points, lines, and polygon feature types were utilized to test the tool as the outputs could easily be visually verified. Because all feature types are converted to buffered polygon features, the toolbox processing was simplified as internally within the toolbox code one would not have to manage points differently than lines, or lines differently than polygons. This decision significantly simplified the Python code in terms of repetitive use of functions regardless of input type.

Development testing of the toolbox software was performed to ensure proper execution. This testing involved repetitive toolbox execution until a successful result was achieved. To aid in testing the overlay of GeoJSON polygon objects from the scene's meta data were utilized to overlay on top of the features to determine if full coverage was achieved. Initial implementations did not successfully cover the entire feature. To address this shortfall, the input feature was buffered by an additional 50 km while calculating the fishnet span of grid points. This resulted in full coverage of the test features by the downloaded imagery and are shown in the RESULTS section to follow.

RESULTS

The Python toolbox consists of a single file named LASST.pyt. This file once created automatically by ArcGIS was further implemented with processing details. To implement this tool the methods `getParameterInfo()` and `execute()` were implemented to accept user input and then when the user clicks the Run button it calls the `execute()` method, respectively. To aid in functional flow additional class methods `getScene()`, `downloadScene()`, and `createBufferedPoints()` were created. This helped with the overall implementation of the `execute()` method. It is in these methods where the use of the `landsatxplore` tool was utilized. The Python toolbox contains approximately 450 lines of code. Reference to the Python code itself on GitHub can be found in Appendix A.

Point Feature Input Results

The first feature that was tested was a point feature type. Figure 6 shows the results of Salt Lake City as a test point input feature. This input feature was composed of five state capitals in terms of timing analysis. However, only Salt Lake City is shown here for simplicity. The tool took 2.1 minutes to download and process 14 scenes for an average of 10 seconds per processed scene. For this test, a buffer length of 50,000 meters was used, date of June 1-1995, and NEAREST scene selection. The tool downloaded two scenes to encompass Salt Lake City. Included in the figure is the buffer shape to verify that the scenes fully bound the user's requested feature with input buffer length.

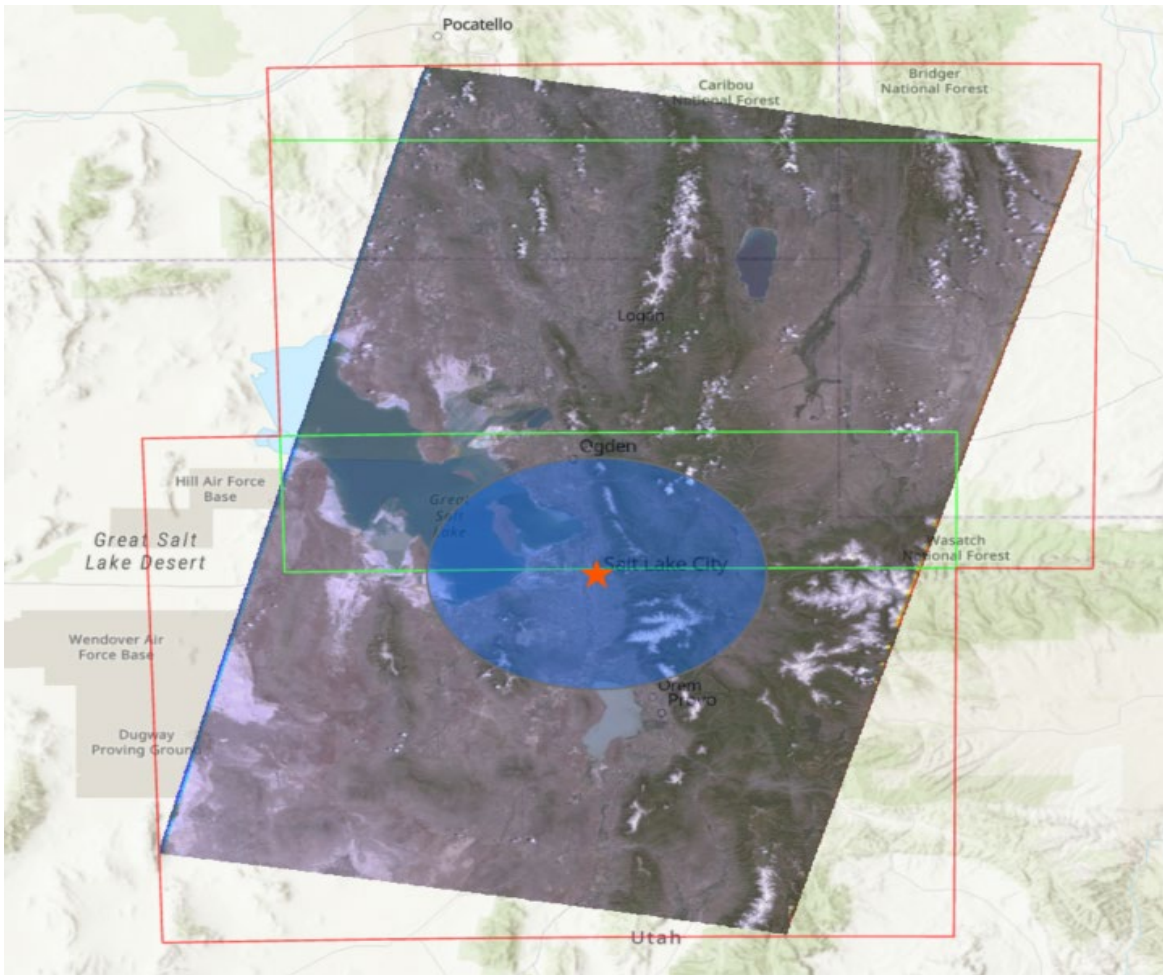


Figure 6. Salt Lake City Mosaic Dataset with 50km Buffer

Line Feature Input Results

The second feature that was tested was a line feature type. Figure 7 shows the results of the Colorado River used as a test line input feature. The tool took about three minutes to download and process 25 scenes for an average of 10 seconds per processed scene. For this test, a buffer length of 50,000 meters was used, date of June 1-1995, and NEAREST scene selection. Included in the figure is the buffer shape to verify that the scenes fully bound the user's requested feature with input buffer length.

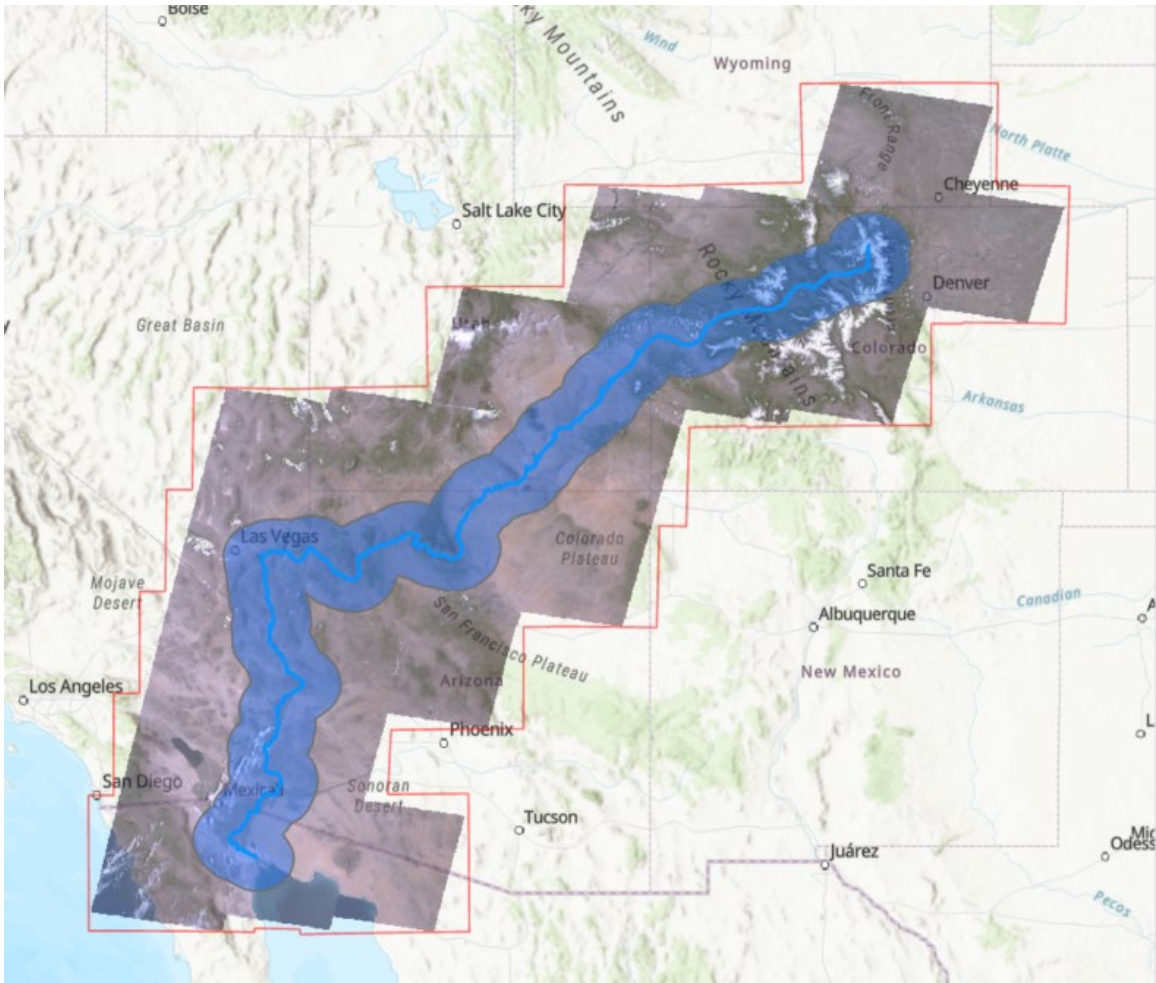


Figure 7. Colorado River Mosaic Dataset with 50km Buffer

Polygon Feature Input Results

The third feature that was tested was a polygon feature type. Figure 8 shows the results of the state of Nevada used as a test polygon input feature. The tool took about 3.1 minutes to download and process 33 scenes for an average of 6 seconds per processed scene. For this test, a buffer length of 50,000 meters, date of June 1-1995, and NEAREST scene selection. Included in the figure is the buffer shape to verify that the scenes fully bound the user's requested feature with input buffer length. The figure

clearly indicates that the state of Nevada was encompassed by the downloaded Landsat scenes.

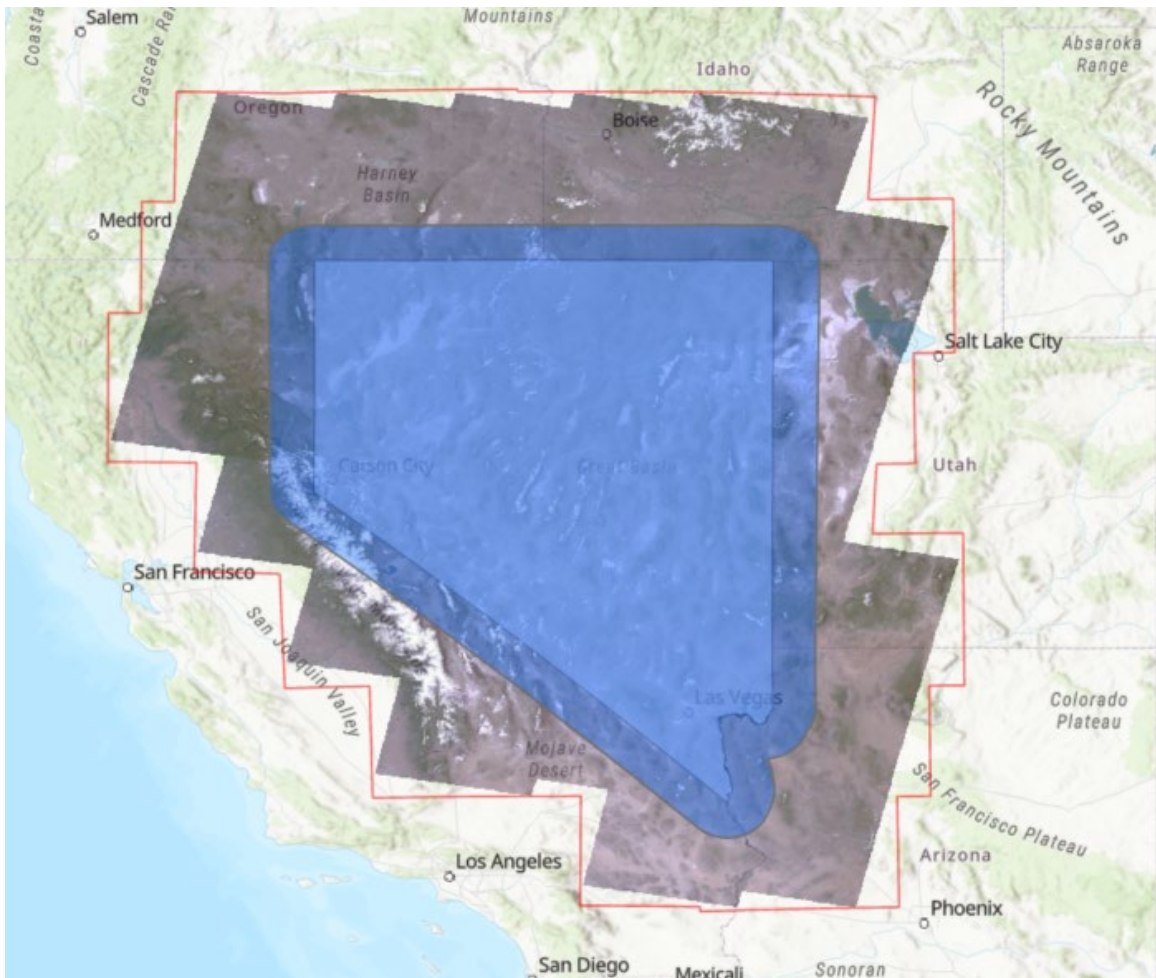


Figure 8. State of Nevada Mosaic Dataset with 50km Buffer

CONCLUSION

The LASST tool is in the form of a Python toolbox when added to ArcGIS Pro, allows the user to utilize the Landsat data easily encompassing the user's feature at the date of interest. The tool makes use of the landsatxplore tool for user interface with the USGS Landsat database but provides a streamlined interface to the end user. The user selects their input feature, date of interest, and max cloud cover percentage, and the tool does the heavy lifting to determine the required bounding scenes and download of those scenes. The result is a Mosaic Dataset of requested Landsat imagery that the user can further manipulate with other geoprocessing tools.

The development and subsequent use of this LASST tool shows a significantly improved speed compared to a manual search and download from the Landsat EarthExplorer web-based interface. For analysts who want fine-tune control over the scenes they would like to utilize for their dates of interest, the LASST tool fits that need with speed and efficiency.

While the tool is quite fast there is some room for further efficiency in the scene sorting and selection process prior to download. A sorting of scenes by percent coverage of the buffered feature could show improvements in overall tool speed. Further improvements to the tool would be helpful, such as an option to convert the data to Top of Atmosphere (TOA) reflectance so that the data can easily be used to support the types of analysis involving reflective indices. This way the data could easily support such tasks as calculated Normalized Burn Ratio (NBR) in the support of forest fire analysis.

Another improvement to the LASST tool would be to remove the dependency upon landsatxplore so that the LASST tool would be stand-alone and not require the

python cloning nor pip install of the landsatxplore tool. This would provide for a much more user-friendly installation and usage paradigm as cloning python environments and pip installs of 3rd party tools can be complicated for some users. A percent dialogue process indicator would be a nice user-friendly feature to be added.

Use of the ArcGIS Python tools and functions takes a bit to get used to. However, one becomes more knowledgeable of the details of the Geoprocessing functions called so often from within the GUI. To call the functions in code lends to a further detailed understanding. Upon the start of the software development the author chose to hard code the inputs to within the execute() method as to focus more upon the scene selection algorithm and less upon user interface. This proved useful to help grasp how to code/develop in ArcGIS so that in the end the user interface could be implemented after the major part of the development, when the developer was more familiar with ArcGIS Python calls.

By making these scenes available in short order, tools like these will help GIS analysts focus on the analysis itself and spend less time performing data management and the searching of online database. This toolbox fills a gap in industry needs and may become a valuable tool in the GIS toolchest.

REFERENCES

- DEPARTMENT OF THE INTERIOR. (1966, September 21). *1966_Udall.pdf*. (N. Release, Ed.) Retrieved April 7, 2024, from [https://d9-wret.s3.us-west-2.amazonaws.com: https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/atoms/files/1966_Udall.pdf](https://d9-wret.s3.us-west-2.amazonaws.com:https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/atoms/files/1966_Udall.pdf)
- EarthExplorer*. (2024, Feb 14). Retrieved April 7, 2024, from USGS EarthExplorer: <https://earthexplorer.usgs.gov/>
- EROS. (2024, Jan 1). *EROS Registration System*. (USGS) Retrieved April 7, 2024, from ers.cr.usgs.gov: https://ers.cr.usgs.gov/login?RET_ADDR=https%3A%2F%2Fm2m.cr.usgs.gov%2Fapi%2Ftest%2Fjson%2F
- ESRI. (2023, November 8). *How to Clone a Python Environment with the Python Command Prompt*. Retrieved April 7, 2024, from Technical Support: <https://support.esri.com/en-us/knowledge-base/how-to-clone-a-python-environment-with-the-python-comma-000020560>
- Forget, Y. (2021, March 1). *yannforget/landsatxplore*, v0.13.0. Retrieved February 14, 2024, from GitHub: <https://github.com/yannforget/landsatxplore>
- NASA. (1981, July 1). *Landsat Data Users NOTES*. NASA, USGS. Sioux Falls, SD: USGS. Retrieved April 7, 2024, from <https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/atoms/files/1981%20July%20Landsat%20Data%20User%20Notes.pdf>
- Team, L. L. (2017). Landsat's Enduring Legacy. In L. L. Team, *Landsat's Enduring Legacy* (p. 345). Bethesda: American Society for Photogrammetry and Remote Sensing. doi:10.14358/ASPRS.1.57083.101.7
- USGS. (2008, April 21). *Imagery for Everyone*. (Technical Announcement) Retrieved April 7, 2024, from [https://d9-wret.s3.us-west-2.amazonaws.com: https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/atoms/files/USGSTechann-20080421-landsat-imagery-release.pdf](https://d9-wret.s3.us-west-2.amazonaws.com:https://d9-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/s3fs-public/atoms/files/USGSTechann-20080421-landsat-imagery-release.pdf)
- USGS. (2023, April 21). *Fifteen Years of Open Data Allows Advancements in Landsat Use and Research*. Retrieved April 7, 2024, from [landsat.gsfc.nasa.gov](https://landsat.gsfc.nasa.gov/article/fifteen-years-of-open-data-allows-advancements-in-landsat-use-and-research/): <https://landsat.gsfc.nasa.gov/article/fifteen-years-of-open-data-allows-advancements-in-landsat-use-and-research/>
- USGS. (2024, February 6). *Landsat Missions Timeline*. Retrieved April 7, 2024, from www.usgs.gov: <https://www.usgs.gov/media/images/landsat-missions-timeline>

APPENDIX A PYTHON TOOLBOX CODE

The python toolbox is available via GitHub at the following URL.
It is available for use/download given an MIT License Agreement.

<https://github.com/toddkeastman/LASST.git>