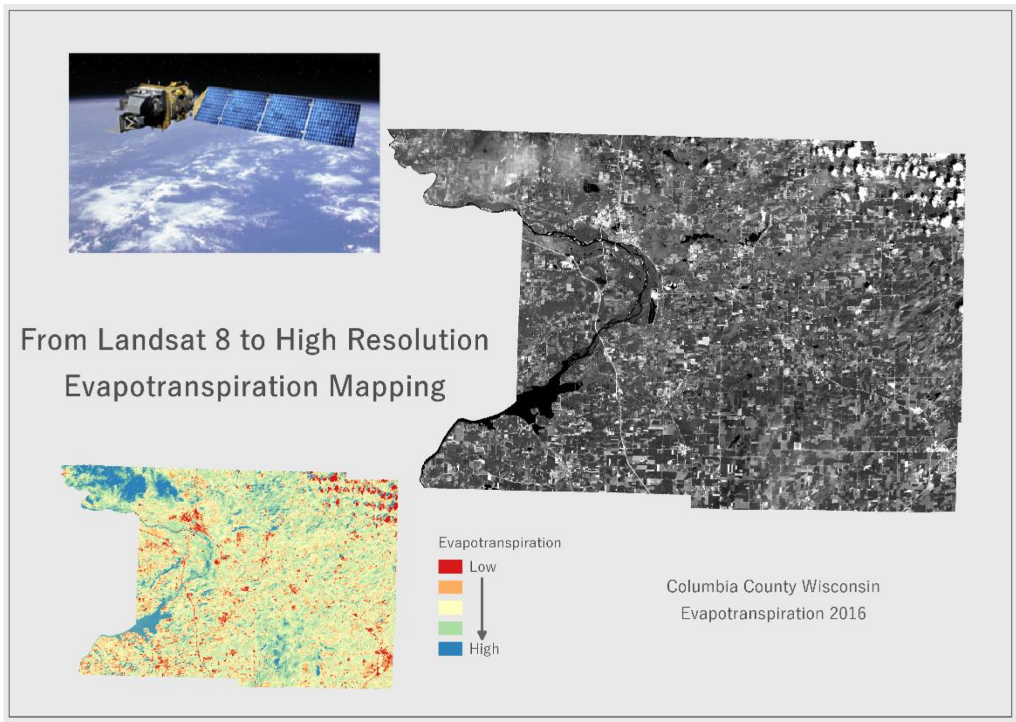
MAPPING EVAPOTRANSPIRATION WITH LANDSAT8 IN THE METRIC MODEL OVER LARGE STUDY AREAS



May 11, 2018

An Investigation of Phase 1

Hydrology models often have limited accurate Evapotranspiration data, causing the calibration to be less accurate. With the METRIC tool, these models could be drastically improved. To make the METRIC outputs more available, this project aims to introduce multi-phase plan and investigation for METRIC Evapotranspiration data outputs to be easily accessible from a web application. Phase 1 is a plan to search, process, and prepare one of the METRIC inputs, Landsat 8 scenes and metadata.

Mapping Evapotranspiration with Landsat8 In the METRIC Model Over Large Study Areas

AN INVESTIGATION OF PHASE 1

Prepared by: Dawson Bancroft-Short
Prepared for: University of Wisconsin Madison GIS Capstone Project GEOG 602
Advised by: Howard Veregin
Submitted to: Brittney Markle
Submitted on: May 11, 2018

CONTENTS

OBJECTIVES

To model evapotranspiration in the METRIC model, there are five different data layers needed. A digital elevation model, landcover data, weather station data, a study area shapefile, and Landsat 8 bands 3-7, 10 and 11 with precisely formatted metadata. Phase 1 of this project is to develop and investigate a tool that easily queries Landsat 8 data available online from a user defined shape or study area, no matter the size, and then processes and downloads the scene(s) within the shape requested. Recommendations and guidance are provided for steps that were unable to be implemented within the time frame of this project.

INTRODUCTION

METRIC -- Mapping Evapotranspiration with Internalized Calibration – is based upon the SEBAL energy balance process developed in the Netherlands by Bastiaanssen (Allen et al., 2009). In the SEBAL process, near-surface temperature gradients are an indexed function of radiometric surface temperature. METRIC is a computer model originally developed by the University of Idaho for mapping calculated evapotranspiration data using Landsat satellite data (Allen et al., 2009). Both SEBAL and METRIC are algorithms that use satellite imagery to solve the earth's energy balance equation. (See **Figures 1**, and **2** below.)

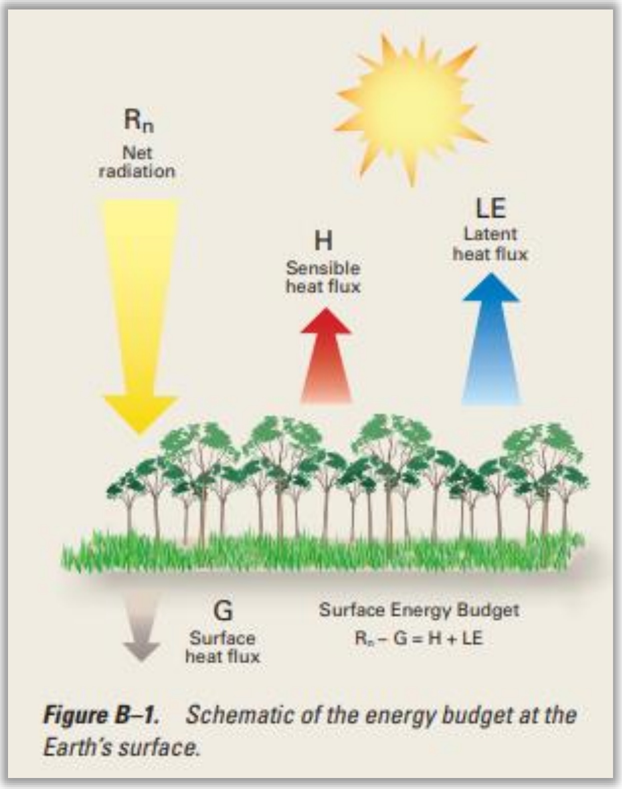
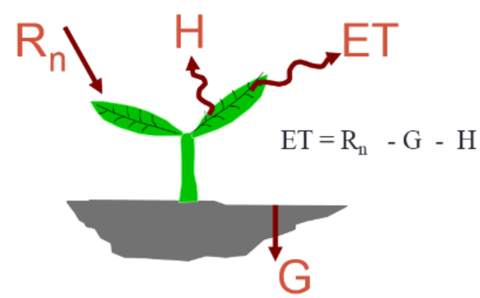


Figure B–1. *Schematic of the energy budget at the Earth's surface.*

Figure 1. (LEFT) Surface Energy Budget. Latent Heat Flux (LE) is the amount of energy needed to change the phase of water from liquid to gas, otherwise known as, evapotranspiration in the METRIC model (Healy et al., 2007).

Figure 2. (BOTTOM) METRIC Evapotranspiration equation represented as energy balance equation (Wikipedia, 2018).



NASA was able to develop an open source python tool for ArcMap that runs the METRIC model and uses a variety of geospatial data as basic inputs: weather data obtained from a nearby weather station, elevation, soil and terrain data, and Landsat 8 satellite imagery (Ely, 2016).

The METRIC tool uses these datasets to produce an evapotranspiration raster image accurately representing crops, vegetation, soil water loss on any specified day of interest that Landsat imagery is available. (Allen et al., 2009) These raster images can have a wide variety of cost-efficient applications and uses for hydrologists, regulators, farmers, and anyone else interested in accurate evapotranspiration data. There are many ways of modeling evapotranspiration (ET) that are both spatial as well as non-spatial (numerical estimates for a region). All have varying levels of accuracy depending on where they are being applied. The METRIC model has performed well in many case studies as well as in a specific comparison analysis between other ET models (Singh et al., 2015).

There are several limitations of the METRIC model in ArcMap as it is currently configured. First, it is designed to be used with a single Landsat image, which limits its utility over large areas (the focus of this project). It is worth mentioning the other two limitations here, as they may be useful to consider while troubleshooting each METRIC limitation individually. These are, that it uses a single weather station, which has limited accuracy over large areas and that it does not include the Priestly-Taylor equation, which is known to improve accuracy of evapotranspiration models and estimations specifically in Wisconsin (personal communication, John Panuska and Scott Sanford, January 19, 2018). Enhancing METRIC to overcome these limitations would enable the model to be a viable source of data for many of the evapotranspiration applications in hydrologic models, climate models, precision agriculture, and beyond. Specifically, with the capability to be applied over large areas (even the entire state), it would be useful for a variety of purposes, including forecasting water needs into the near future as well as modeling the effects of climate change on water in the state of Wisconsin and beyond.

To visualize the slight differences that may occur when modeling groundwater flow, a very basic model was done using MODFLOW in QGIS FREEWAT, with and without a METRIC ET raster. (See **Figures 3** and **4** below) While the differences are slight, the purpose of this quick procedure was to simply show that there *are* in fact differences. Since this model was not the focus of this project, it should not be considered highly calibrated, but more of a visual reference for the discrepancies that can occur in hydrologic models of complex inputs and outputs designed to be spatially analyzed.

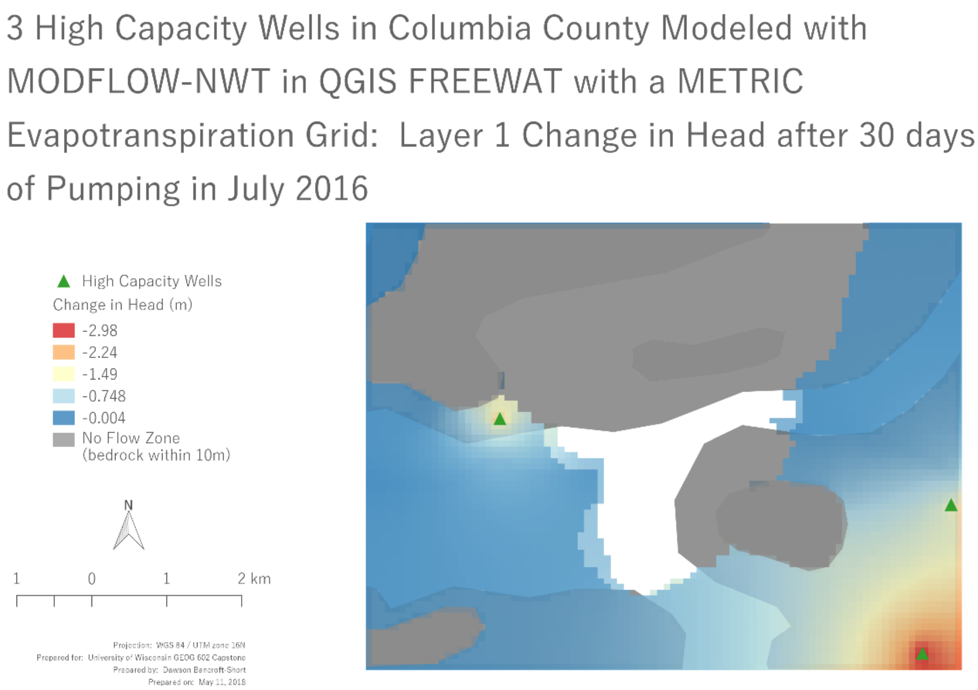


Figure 3. METRIC Evapotranspiration raster output (see **Figure 7** of METRIC outputs in Appendix) used in a MODFLOW groundwater flow model for the change in heads during the month of July 2016.

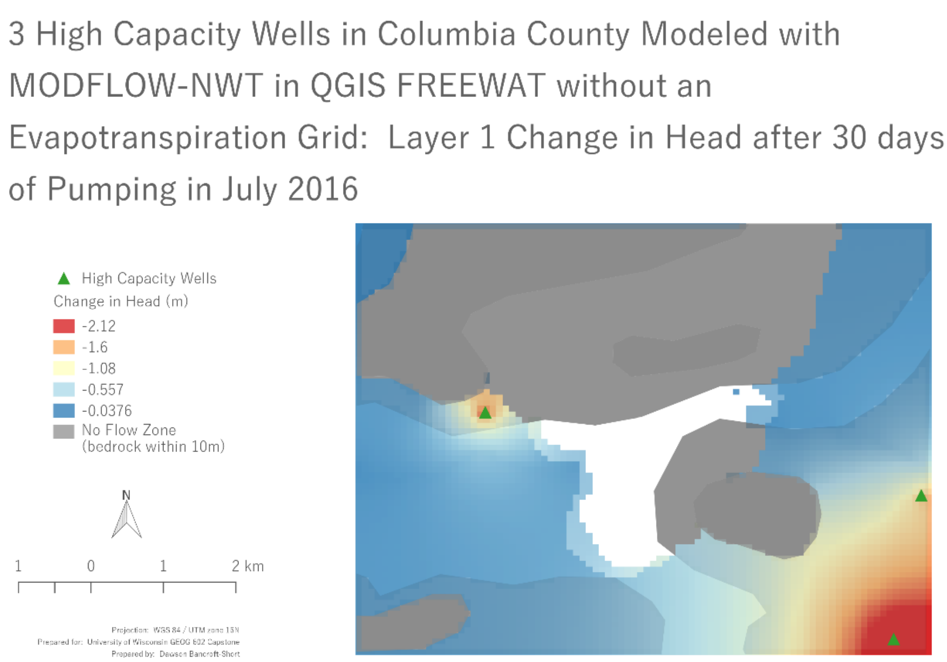


Figure 4. MODFLOW output without an evapotranspiration model layer input in July 2016.

As discussed in the proposal previously submitted for this project, the enhancement of the capabilities to use Landsat over large study areas was chosen as phase 1, not only because of time constraints, but it also has a range of applications outside of the METRIC model. Phases to implement an easily accessible tool or source for METRIC output data sets to follow, are merely recommendations based on the investigations of this project and should be continuously re-evaluated. Using Landsat over large study areas can be a tedious task, often requiring large amounts of storage space and manual processing steps that is burdensome in applications that only needs to temporarily handle such data to produce final data outputs. The second, and slightly more complex challenge regarding Landsat in the METRIC model, is the need for a very specific format and content of a metadata file (.mtl). METRIC will not successfully run without the specific data it needs from Landsat metadata that is typically provided with single scene downloads. This issue is a bit more complex, as there tends to be a certain amount of data lost with each analysis process applied to raster images, such as Landsat scenes (Flasher & Becker, 2017). That's not to say it's impossible, but it does take a certain amount of understanding of the computer program, as well as the specific calibrations of the METRIC evapotranspiration model. In the end, a behind the scenes function that can access Landsat data triggered by a user query of a specific area of interest or time, completes phase 1. Implementing such a function is unfortunately beyond the scope of this paper due to limited time and resources. The trials and errors of attempting each of the methods investigated are as much a part of the process as they are the results. For phase 1, we hope to give insight to anyone interested in not only evapotranspiration mapping, but also those with those who wish to a) use Landsat imagery over large study areas, b) process Landsat imagery for temporary computing in a cloud environment to avoid local storage "clutter", and c) create new metadata from post-processing of scenes that can comply with any of the applications that use strict supplemental information and formatting.

METHODS AND CONCEPTUALIZATION

The investigation for this project began with a look into the different ways of downloading and processing Landsat that could eventually be queried with vectors that stretch across multiple scenes in on area of interest. A common source for Landsat imagery is United States Geological Survey (USGS) and is generally a good standard for those looking for one or two scenes and have the time and resources for manually searching, requesting, some pre-processing, and downloading them. USGS does in fact offer an API through the Earth Resources Observation and Science (EROS) Center Science Processing Architecture (ESPA) for querying their Landsat libraries and downloading bulk orders, but we found it to have too many limitations for what we needed (USGS, 2018). Never the less, these limitations were unknown until attempting to execute a program that used that API, so for a short while it still appeared to be a good place to start. While there may have been more capabilities of this option, the choice to deem it unviable for this project was a simple one in that; it did not appear to be any more useful than using the web application manually. When you download Landsat from almost any of the four USGS applications (EarthExplorer, GloVis, or LandsatLook Viewer), you usually fill up a “shopping” cart as you browse with the scenes and datasets you are interested in. When you are ready to download from that cart, you then must *request* to download them and then wait for a link to be sent to an email address typically associated with a pre-registered account with access credentials. There is an unspecified lapse in time for when they arrive to your inbox, and the link still needs to be physically clicked to begin the actual download process.

At this point in the investigation, it was clear that other options needed to be explored and that lead to Amazon Web Service (AWS). As of March 19, 2015, Amazon Simple Storage Service (Amazon S3) announced Landsat on AWS as part of an initiative to expand access to the increasing research demand for computing data in the cloud in areas such as climate change, agriculture and space exploration (Amazon, 2016). In just one year, Amazon boasts logging over 1 billion requests for Landsat 8 data from the S3 cloud. Amazon S3 has become a powerful tool for many other data storage demands across the world and industries alike. It is not only a viable option for preparing Landsat 8 data for the METRIC model but could very well be the place that all phases of this project process, download, and even run the METRIC model. From the S3 cloud, it could then provide evapotranspiration data for say, a static web page where users could go and simply provide an area of interest for ET data and have years of outputs to choose from (as far back as Landsat 8 data is available). One of the core initiatives of AWS is to provide open data storage to both commercial customers and public-sector customers in a scalable manner on the cloud (Flasher & Becker, 2017). Specifically, they aim to remove what they refer to as "undifferentiated heavy lifting", or what is often 80% of the data preparation work in data science (Flasher & Becker, 2017). This translates directly to geospatial data and especially satellite data. Below in **Figure 4** is a screenshot from a talk given at AWS re:INVENT by Joe Flasher in 2017 where Drew Bollinger at Development Seed graphs the time difference in downloading traditional "tarball" or archived Landsat data and unarchived AWS Landsat data.

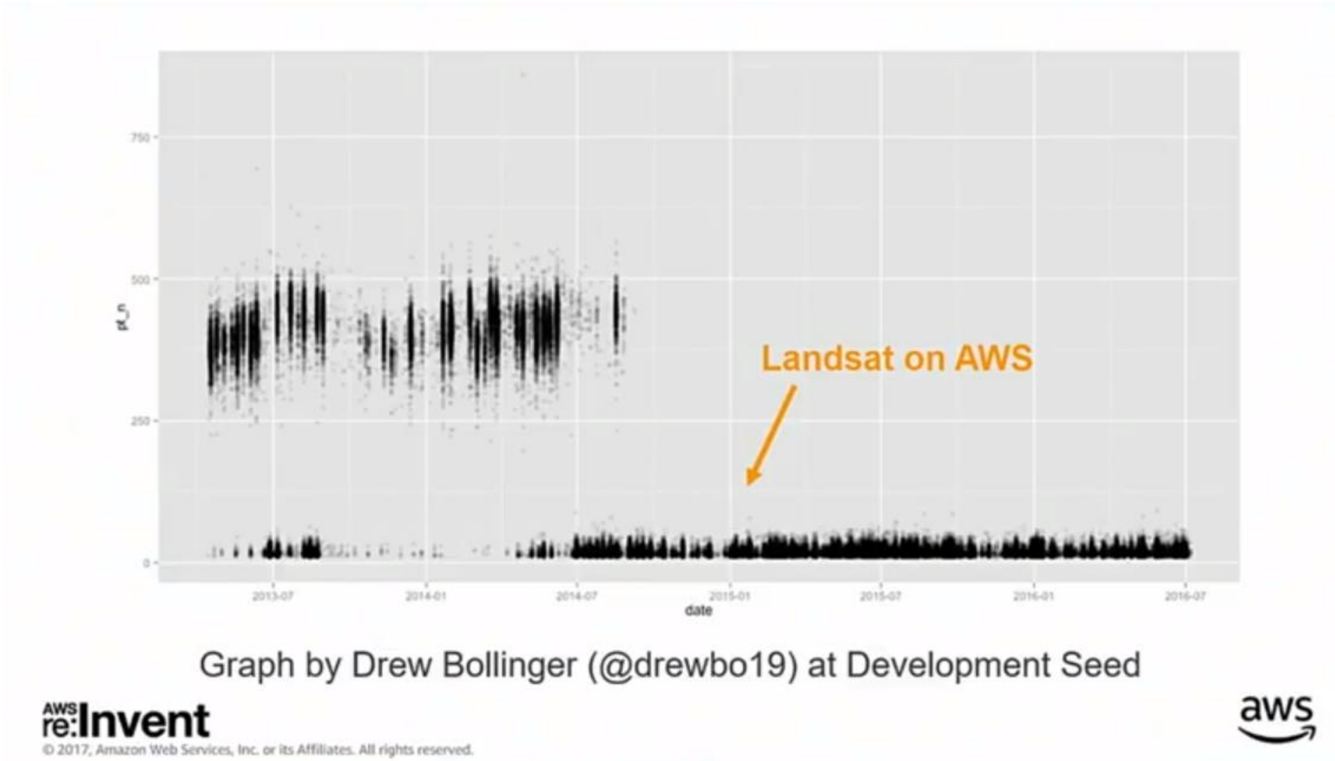


Figure 5. Landsat downloading time differences between traditional archived data and AWS unarchived data. The left is downloading single scenes within a couple of minutes, and the left in a matter of seconds (Flasher & Becker, 2017).

Further use of the AWS platform also appears to also be a viable option for further phases of the METRIC model project because one year after launching Landsat on AWS in September of 2016, Amazon launched Earth on AWS for “building planetary-scale application in the cloud with open geospatial data” (Amazon, 2018). The other data sets available from Earth on AWS are currently: NAIP 1m Imagery, Terrain Tils, GDELT – A global Database of Society, NEXRAD Weather Radar, SpaceNet Machine Learning Corpus, and NASA Earth Exchange. Since the other data layers needed to run the METRIC model include: a digital elevation model (DEM), landcover, and weather station data, working within the S3 environment appears to be another valuable option (Barr, 2018). While AWS is well documented and someone with moderate development skills could write custom programs for interacting with the S3 cloud, some other credible organizations have taken the liberty of doing so. Many of these tools for processing in the cloud would work for accomplishing exactly (or very close) to what is needed for phase 1 of this project. Discovery of these options continued until the very end of this project deadline and has likely only scratched the surface of true possibilities.

The exploration of the pre-developed tools mentioned above for implementing phase 1 began with Landsat-util. Landsat-util was the first of what is now many of the libraries developed by Development Seed for interacting with AWS S3 (Development Seed, 2014). It works as a command line utility available on GitHub and according to Development Seed, does three things well: "searches and loads Landsat metadata, makes downloading easier, and it processes the data, with natural color-correction and pansharpening, and gets it ready for use in Mapbox Studio or tool of your choice" (Development Seed, 2014). What appeared to be an ideal first solution, was eventually concluded that there were some major dependencies that didn't make it as viable as previously conceived. This, as with most ideas discussed here, took some time working through. The first, and likely most difficult obstacle, was that it was not configured to run on Windows. It was for use in a Mac OS, Ubuntu, or Docker environment. Due to many GIS systems running in Windows environment, there needed to a work-around method to make this a viable solution. While the Ubuntu community is large and well documented, the average user may not be familiar with this operating system, but that was not a *completely* limiting factor just yet. Many Windows environments will allow a dual operating system with Ubuntu, so this began the next steps of investigating how to make this possible. Windows 10 specifically offered what seemed to be an easy solution with their new Ubuntu application that could be used to run a bash shell to develop within the Ubuntu operating system environment with little extra effort to install and start using immediately. Landsat-util was then successfully installed in the Ubuntu environment and from there Landsat scenes were able to be downloaded. There were still many limitations to this solution, such as little pre-processing could be done, as well as the search and download function only successfully running on scenes that were used in the examples within the library documentation. While this very well could have been overcome with more time and resources, the next discovery made that a less efficient use of time.

Development Seed had already addressed the limitations mentioned above with what was soon discovered for this project, and that solution was Sat-utils. One of the main points referenced by Matt Hanson from Development Seed presenting as FOSS4G was that Landsat-util was never developed for heavy scientific processing capabilities (Hanson & Jazayeri, 2016). So, while it may have been a good option for a portion of our project needs, eventually the potential for the development for this project to get more complex, made looking into the new Sat-utils libraries a better choice (Bollinger, Flasher, & Jazayeri, 2018). Sat-utils is a collection of smaller libraries, each with specific functions for querying, processing, and downloading Landsat 8 and Sentinel 2 satellite scenes. While seemingly similar to Landsat-util on the surface, it works much differently "under the hood."

Sat-utils libraries include: sat-testdata, sat-process, sat-fetch, sat-api-lib, Landsat8-metadata, sat-stats, sat-testdata2, and Landsat8-util. Some basic implementation attempts of phase 1 within this tool was enough to strongly recommend it's use in the completion of phase 1. The choice of what to try and implement in the scope of this project timeline was limited by the amount of time allotted for this investigation. Therefore, the start of the program for interacting with the AWS S3 cloud in a CLI (command line interface) uses the sat-search and sat-download libraries as a place to start and familiarize with the S3 architecture. Furthermore, with more time, a strong team of developers may consider starting with the full sat-api-lib or some of the other more specific libraries later.

The fourth and final discovery of this investigation was unfortunately made at the very end of the time allotted on this part of the project and very well may be the most suitable solution so far. The release of ArcMap 10.5 by ESRI included a new feature in their Image Server platform for "scalable raster analysis and image processing", that now utilizes AWS cloud as a processing environment (Muller, 2018 and Flasher & Becker, 2017). The scope of this project would likely benefit highly from this feature for many reasons, but does have a rather large limitation as well, and that is that ESRI only offers their raster analytics to their enterprise accounts. While it could be likely that an organization implementing this project would have and enterprise account, and the METRIC tool itself needs at least an ArcGIS Desktop Personal Use license with a spatial analyst toolbox, it is *not* a viable solution for those who do not. Despite this being the case, investigation of this route should still strongly be considered.

After reviewing all of the Landsat processing and download options to the extent that time would allow, the best recommendations this investigation can give would be to utilize Sat-utils, *unless* you have access to an ESRI ArcGIS Enterprise account. Specifically, within the Sat-utils libraries, the functions that seem the most capable of completing phase 1 are, the sat-search, sat-download, sat-testdata, and Landsat8-metadata. The first function would be to search for the desired area of interest that would also satisfy the dates of interest and this is something that sat-search is capable of. The next would be to utilize the sat-tesdata library's capabilities of querying vector regions that require multiple scenes and mosaicking them together in preparation for download. The third step would be to generate Landsat8 metadata for the newly processed scene of interest before finally downloading with the sat-download library.

While the suggestions above seem easy enough in theory, there may be better ways of utilizing the Sat-utils API and all development for such a program should be given sufficient amounts of time in preparation for things to not always work as they should. Dependencies can change between libraries, and as with all software, updates and version changes are inevitable. One of the main benefits of the Sat-utils library, is the developers at Development Seed seem to be capable of keeping up with most of these issues.

Method	Developer	Pros	Cons
EROS ESPA Bulk Download API	USGS	1. Secure 2. Decent documentation 3. Relatively easy to implement with little programming experience.	1. Is not capable of a lot of complex processing 2. Data is still archived (or "tarballed") causing bulk downloads still take a lot of time 3. Manual step between requesting scenes and downloading them from a link sent to an email address.
Landsat-util	Development Seed	1. Decent amount of documentation 2. Has been around long enough to have enough users who have asked about/solved issues they may have faced 3. Is good at the few things it does	1. Does not run on Windows 2. Was never devloped for heavy scientific processing capabilities 3. With enough similarities to Sat-utils while still being different, may as well use the later.
Sat-utils	Development Seed	1. Many libraries to choose from 2. Functions of each library differ vastly in simplicities and complexities 3. There are applications that use the sat-api-lib in live websites, which encourages developers to keep everything up to date 4. With more time and resources, is fully capable of completing phase 1 of the METRIC over lagre study areas project	1. Requires moderate to advance programing skills 2. Some of the libraries have conflicting dependancies at the time of this project (i.e. numpy version 1.XXX for one thing and numpy version 2.XXX for another)
Raster Analytics	ESRI	1. Works within a very familiar GIS GUI that is widely used 2. Utilizes AWS S3 Cloud computing environment 3. Could potentially be the best choice for phase 1 of this project AND all phases to follow	1. Not everyone has access to an ArcGIS Enterprise liscence 2. Maintaining access to an enterprise license for long periods of time requires development in this environment to be with an organization that can afford to keep it running long-term

Table 1. A compiled review of investigation results and suggestions.

RESULTS AND DISCUSSION

As stated in the project proposal submitted in January, this project aimed to investigate possible solutions to limitation within the METRIC python tool developed by NASA for use in ArcMap. Implementation of these solutions was and always will be time dependent, but the need for numerically and spatially accurate evapotranspiration data is and will continue to grow with the demand for the earth's water resources to be used and regulated more efficiently. Furthermore, Landsat imagery continues to be a valuable source for many other types of analysis, making phase 1 an important step within this project and beyond. While being a powerful tool as is, METRIC models from the tool for ArcMap still has its own limitations. Looking back at the limitations discovered for this project, they were: the ability to only run the METRIC model with a single Landsat scene *with* the specific metadata file needed for accurate tool calibrations, the limitations of a single station weather data file, and the varying ways of calculating ET, with one of the best for Wisconsin being the Preistly-Taylor equation.

Implementing these solutions was never fully realized, and as mentioned earlier, the nature of these investigations was to endure a lot of trials and errors. What this project has produced, is months of evaluation that hopes to give others the resources necessary to not have to spend time on some of the trials, as well as what we believe to be the best path to take moving forward for pre-processing Landsat imagery for use in the METRIC model. Each individual requirement for querying, requesting, and processing of Landsat scenes appears to be fully capable within the Sat-utils library, even if the way of utilizing them could done in many ways.

FUTURE EFFORTS AND CONCLUSION

Future efforts of this project should gear towards utilization of the AWS S3 cloud computing environment alongside the Sat-utils API, and very likely the Earth on AWS data collections that Amazon has also released.

The following phases should be looked at in an order that allows for METRIC models to be easily accessible in the least amount of time possible. What this means is that while adjusting the tool to incorporate the Priestly-Taylor ET equation and utilize more spatially accurate weather data (from single point to either multiple points or raster format), the most import aspects of a METRIC model web application or browsing tool would be; the ability to quickly process bulk amounts of Landsat images, as well as the DEMs, and land cover datasets that allow the METRIC model to run as is. From there, calibrating other areas of the tool could come over time. Also, the replication of the tool to run in an open source GIS system (such as QGIS), would be another useful idea for making and keeping ET data available in the future.

ACKNOWLEDGMENTS

I would like to thank my advisor, Howard Veregin and the University of Wisconsin Madison for the opportunity to pursue this project and the resources to do so. I would also like to thank Jim Giglierano the Wisconsin GIS Officer, and Scott Sanford and John Panuska from the UW Madison Extension office for taking the time to meet and encourage me to begin this project.

REFERENCES

Alam, M., & Rogers, D. H. (2009). Irrigation Management for Alfalfa – KSRE Bookstore. Retrieved from: https://www.bookstore.ksre.ksu.edu/pubs/MF2868.pdf

Allen, J., Tasumi, M., Trezza, R., Robison, C., Kjaersgaard, J., Morse, A., . . . Wright, J. (2009). METRIC: High Resolution Satellite Quantification of Evapotranspiration. Lecture presented at Powerpoint Presented Online in University of Idaho. Retrieved May 11, 2018, from http://wdl.water.ca.gov/wateruseefficiency/sb7/docs/Metric_Training_Presentation_Part_one_Boise.pdf

Amazon. (2016, January 11). Landsat on AWS: Half a Year, Half a Billion Requests | Amazon Web Services. Retrieved May 1, 2018, from https://aws.amazon.com/blogs/publicsector/landsat-on-aws/

Amazon. (2018). Earth on AWS. Retrieved from https://aws.amazon.com/earth/

Barr, J. (2018, March 20). Earth on AWS: A Home for Geospatial Data on AWS | Amazon Web Services. Retrieved from https://aws.amazon.com/blogs/aws/earth-on-aws-a-home-for-geospatial-data-on-aws/

Bollinger, D., Flasher, J., & Jazayeri, A. (2018). Sat-utils. Retrieved from https://github.com/sat-utils

Development Seed. (2014, August 29). Power tools for Satellite Imagery. Retrieved February 2, 2018, from https://www.developmentseed.org/blog/2014/08/29/landsat-util/

Ely, J. (2016, April 22). NASA-DEVELOP/METRIC. Retrieved January 23, 2018, from https://github.com/NASA-DEVELOP/METRIC/tree/arcpysa-import-fix

Flasher, J., & Becker, P. (2017, December 01). AWS re:Invent 2017: How Esri Optimizes Massive Image Archives for Analytics in the C (ABD402). Retrieved from https://www.youtube.com/watch?v=U486YxIDoeM

Hanson, M., & Jazayeri, A. (2016). Sat-utils: Landsat, Sentinel and the use of open raster data. Retrieved from https://av.tib.eu/media/20399 FOSS4G Bonn 2016

Healy, R.W., Winter, T.C., LaBaugh, J.W., and Franke, O.L.. (2007). Water budgets: Foundations for effective waterresources and environmental management: U.S. Geological Survey Circular 1308, 90 p. Retrieved on May 1, 2018 from https://water.usgs.gov/watercensus/AdHocComm/Background/WaterBudgets-FoundationsforEffectiveWater-ResourcesandEnvironmentalManagement.pdf

Panuska, J. C. (January 19, 2018). Interviewed by D. A. Bancroft-Short [In person]. University of Wisconsin Biological Systems Engineering, UW Extension. Madison, WI.

Sanford, S. A. (January 19, 2018). Interviewed by D. A. Bancroft-Short [In person]. University of Wisconsin Biological Systems Engineering, UW Extension, Rural Energy Program. Madison, WI.

Singh, S. K., & Irmak, A. (2011). Treatment of anchor pixels in the METRIC model for improved estimation of sensible and latent heat fluxes. Hydrological Sciences Journal 56(5), 895-906.

Singh, S. K. & Senay, G. (2015). Comparison of Four Different Energy Balance Models for Estimating Evapotranspiration in the Midwestern United States. Water, 8(19), 1-19.

Wikipedia. (2018, April 03). METRIC. Retrieved May 1, 2018, from https://en.wikipedia.org/wiki/METRIC

APPENDIX

```python
from subprocess import Popen, PIPE, STDOUT
from sdownloader import Landsat8
from tempfile import mkdtemp
import sys
import requests
import json
import urllib2

# Test Shape and download directories etc....
# sat-search --intersects D:/Libraries/Documents/School/CLASSES/Spring2018/CAPSTONE/DATA/SampleShapes/portagecounty.geojson --date 2016-07-01 --printmd


# --datadir D:/Libraries/Documents/School/CLASSES/Spring2018/CAPSTONE/DATA/Sat_Utils/Sat_Search/

def sat_download():
    folder = ''
    while folder == '':
        folder = raw_input("Enter the folder to download scenes: ")
        if folder != '':
            l = Landsat8(download_dir=folder)
            scenes = l.download([scene_ids])
            print(scenes)
            [s.name for s in scenes]
            {s.name: s.files for s in scenes}
        else:
            print("Something went wrong.")
            quit()

def sat_scenes():
    global scene_ids
    scene_ids = ''
    while scene_ids == '':
        scene_ids = raw_input("Choose the scenes you would like to download in single quotes separated by a comma: ")
        if scene_ids != '':
            return scene_ids
        else:
            print("Something went wrong.")

def sat_search():
    # file = raw_input("\nPlease enter the complete geojson file path to the shape or region of interest: ")
    # cmd = "sat-search --intersects"+file+"--printmd"
    # cmd = "sat-search --intersects D:/Libraries/Documents/School/CLASSES/Spring2018/CAPSTONE/DATA/SampleShapes/portagecounty.geojson --date 2016-07-01 --printmd"

    # Invalid Scene for portage county, trying date and path row instead
    cmd = "sat-search --satellite_name landsat --param path=24 row=30 --date 2016-07-01 --printmd"
    p = Popen(cmd, stdout=PIPE, stderr=PIPE)
    stdout, stderr = p.communicate()
    results = stdout
    print results
    # while results != '':
    #   print results
    #   if results == '':
    #       print("Sorry, there were no scenes for that region.")
    #   else:
    #       print("Something went wrong, sorry.")
    #       quit()
sat_search()
sat_scenes()
sat_download()
```

Figure 6. The START of a basic implementation for a command line tool that queries the AWS S3 cloud and downloads Landsat 8 scenes. Currently it ends on an error about the availability of scenes, but could be a place to start the implementation of phase 1.

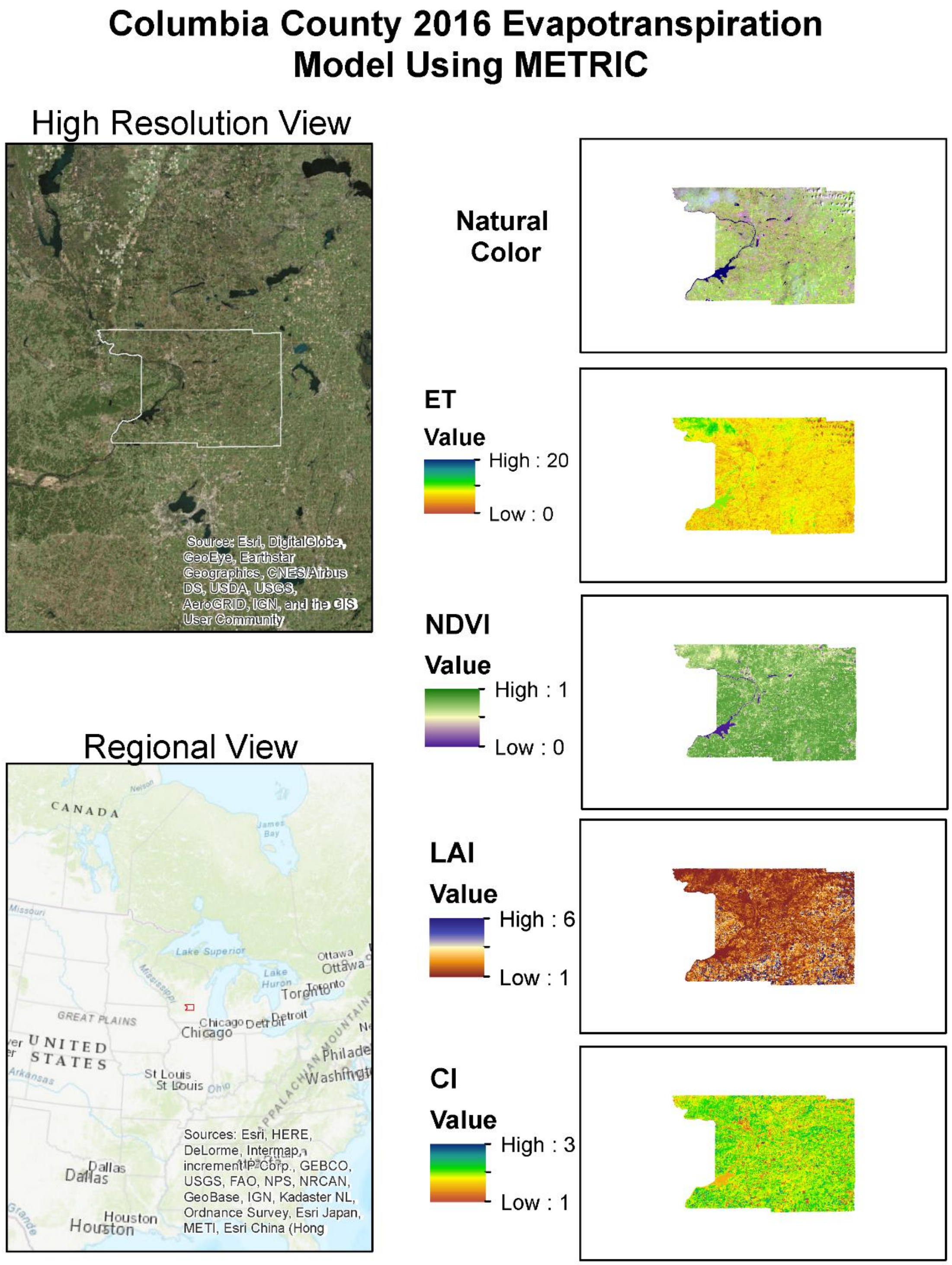


Figure 7. METRIC raster outputs from METRIC tool developed for ArcMap.